CAPITAL UNIVERSITY OF SCIENCE AND TECHNOLOGY, ISLAMABAD

Comprehensive Intrusion Detection System Over Edge Computing

by

Arslan Akram

A thesis submitted in partial fulfillment for the degree of Master of Science

in the
Faculty of Computing
Department of Computer Science

2022

In honor of my parents, supervisor, and all my teachers, I dedicate my dissertation. My father, who is the most helping man I have ever known, deserves a special feeling of gratitude.

CERTIFICATE OF APPROVAL

Comprehensive Intrusion Detection System Over Edge Computing

by

Arslan Akram

(MCS203029)

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Arif Jamal	FU, Islamabad
(b)	Internal Examiner	Dr. Nayyer Masood	CUST, Islamabad
(c)	Supervisor	Dr. Masroor Ahmed	CUST, Islamabad

Dr. Masroor Ahmed
Thesis Supervisor
December, 2022

Dr. Abdul. Basit Siddiqui
Head
Dept. of Computer Science
December, 2022

Dr. M. Abdul Qadir
Dean
Faculty of Computing
December, 2022

Author's Declaration

I, **Arslan Akram** hereby state that my MS thesis titled "**Comprehensive Intrusion Detection System Over Edge Computing**" is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

Arslan Akram
Registration No: MCS203029

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled "**Comprehensive Intrusion Detection System Over Edge Computing**" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

Arslan Akram
Registration No: MCS203029

Acknowledgement

As a first step, I would like to acknowledge Allah Almighty, the Omnipotent, the Omnipresent, the Most Kind and the Most Merciful, who has blessed and helped me throughout my studies and granted me success in every field of my life. Next, distinct praise goes to Hazrat Muhammad (peace and blessings of Allah be upon him), who is the torch of guidance for humanity throughout eternity. Throughout the thesis process, I am grateful for the guidance, inspiration, and mentorship provided by my supervisor, Dr. Masroor Ahmed.

He has been a regular source of support and assistance to me. It would not have been possible to complete this thesis without his supervision and mentorship. Throughout my academic career, I cannot sufficently express my gratitude to my dear parents, my brothers, and my friends for their motivation, encouragement, and prayers. Finally, I wish to thank all the professors from the Computer Science Department of the Capital University of Science and Technology, Islamabad for providing me with knowledge in a professional setting.

Arslan Akram

Abstract

In modern world, most work can be done by using computer systems that make their work easy and accurate. As industries automate their work with IoT, they enhance its performance, but these changes generate some security flaws which could compromise system security and lead to system damage or service disruptions. Nowadays, war tactics have become more complex, and countries launch cyber attacks on their enemy countries to cause damage to their resources. Therefore, we need security solutions that help defenders prevent and detect cyber attacks in time. Due to the vast use of IoT in every industry, we need comprehensive security solutions to detect attacks accurately. Proposed research covers the intrusion detection system which has multiple components and uses machine learning to detect modern attacks with maximum number of attacks classes and network protocols. Other components of proposed Intrusion Detection System are based on signature based and connected threat mitigation which helps to mitigate the attack automatically. This research proposed the deployment architecture for proposed IDS, with selection of best possible dataset and Machine Learning algorithm to use in anomaly based engine, keep an eye and track on network used in sensor intrusion detection engine. The research basically enhances previous work that uses layerd machine learning to detect attacks. By using the proposed comprehensive IDS, this research has produced a lightweight IDS that can be used over edge servers to detect attacks in real time. The future work of research will cover testing of the proposed system on low-resource servers.

Contents

List of Figures

List of Tables

Abbreviations

ACL	Access Control List
ACMS	Access Control Management Systems
Af1	Accuracy (f1-score)
APT	Advance Persistent Threat
DoS	Denial of Service
DT	Decision Tree
EDR	Endpoint Detection and Respond
FW	Firewall
HIDS	Host Based Intrusion Detection System
IDS	Intrusion Detection System
IIoT	Industrial Internet of Things
IoC	Indicator of Compromise
IoT	Internet of Things
IPS	Intrusion Prevention System
LR	Logistic Regression
MAf1	Macro Avg (f1-score)
MAP	Macro Avg Precision
MAR	Macro Avg Recall
MitM	Man in the Middle
ML	Machine Learning
MLP	Multi-Layer Perceptron
NB	Naïve Bays
NIDS	Network Intrusion Detection System
NMS	Network Management System

NN	Neural Networks
pABC	parallel Artificial Bee Colony
RF	Random Forest
SIDS	Sensor Intrusion Detection System
SIEM	Security Incident and Event Management System
SVM	Support Vector Machine
TTs	Time Taken in seconds
WAf1	Weighted Avg (f1-score)
WAP	Weighted Avg Precision
WAR	Weighted Avg Recall
XDR	Extended Detection and Respond
XSS	Cross Site scripting

Chapter 1

Introduction

1.1 Background

Computer Technology influences humans' lifestyles and improves their standard of living. Computers play a vital role in every field of life, including industry, healthcare, defense, etc. Computer technology promotes automation, making things easier and more efficient.

The emergence of IoT (Internet of Things) improves daily life through automation and provides control over computers. IoT is the network of interconnected devices that include sensors, actuators, computers, software, and networks. Sensors sense and transmit data over the network to the central computer where software processes it and sends the signals to actuators to perform certain actions based on conditions. IoT has very rich applications in daily life like smart cities, self-driven cars, IoT-based farming, wearables, smart industry, smart health care, smart grids, smart traffic management, etc.

IoT works over the network which raises some potential security threats. If an attacker compromises the vulnerabilities in design architecture or in technology, the attacker could be able to control the system by gaining system access or affect it in different ways. To overcome the security threats, we need some sort of solution that helps the system administrator to detect and prevent the system from cyber-attacks. Researchers continually research in this field to find mechanisms that increase security and overcome threats.

Cyber Security is the evolving field and sub-branch of information security. Cyber Security is basically the protection of digital assets to protect the hardware, software, and information. Security can be achieved by correcting system architecture and the use of some sort of tool that helps the administrator to detect digital attacks and then add preventions for specific attacks.

1.2 Introduction

The evolution of computers, networks, and technology has completely changed the way we live. Computers automate most of the work and make it easier, efficient, and effective. With computers, we can communicate in every location where we have satellite signals or physical networks. With the computer, IoT also improves the daily application of life in our respective fields and enables us to automate the systems or control the systems from a computer or mobile device in local or remote locations using network and cloud-based technologies. A number of security issues arise as a result, which raises some questions about the use of the system. The researchers investigate the security issue and develop ways to increase the security and protect the system in order to make it usable.

Due to automation and smart systems, IoT devices are increasing in number, which increases the load on networks and clouds. IoT Devices send their data to the cloud where an application is hosted and in response, applications return instructions to perform the specific actions. There are some applications that require real-time responses, but due to network and cloud latency, real-time decisions are not possible. Researchers propose Edge Computing as a means of addressing such problems. The basic idea of edge computing is to employ a hierarchy of edge servers with increasing computation capabilities to handle mobile and heterogeneous computation tasks offloaded by the low-end IoT devices, namely, edge devices. Edge computing has the potential to provide location aware, bandwidth sufficient, real-time, privacy-savvy, and low-cost services to support emerging IoT applications. According to the latest report by Statista, the market size of edge computing in the United States was projected to reach $1031millionby2025fromthecurrent84.3$ million in 2018 [1].

As IoT devices have very limited memory and processing power, they are vulnerable to different attacks over the network, which could adversely affect their useability. Those devices or the entire system may be subject to different cyber attacks. Currently, we also face a threat of 5th Generation warfare, which involves cyber-attacks to disrupt services. Other countries' automated systems are damaged or interrupted by attackers. In order to make applications and systems more reliable and safer, we need to make them smart and intelligent so that they can detect attacks and make the system more reliable. Due to the complexity of applications, development teams ignore advanced security measures. Therefore, we need to use third-party tools to keep an eye on the system. It is good for detecting attacks using third-party systems in enterprise organizations, but not for small systems like home automation systems, because normal humans are not familiar with security tools. Thus, we should provide some security and attack detection features in those systems instead of really relying on third parties.

System administrators have access to a wide range of tools for ensuring proper security, but each tool increases budgets. The most common tools are SIEM(Security Incident and Event Managment System), EDR(Endpoint Detection and Respond), XDR(Extended Detection and Respond), NMS(Network Managment Systems), IDS(Intrusion Detection System), IPS(Intrusion Prevention System), Firewall etc. Each solution has its own capability to ensure the security level at different layers. These tools provide layered security. Our topic is IDS which works at different layers and detects intrusions.

Intrusion Detection Systems detect intrusions of different types like HIDS(Host Based Intrusion Detection System), NIDS(Network Based Intrusion Detection System) etc. A HIDS is installed at an endpoint to ensure host security, while a NIDS is installed at the network layer to detect intrusions over the network. In order to build an IDS, two approaches are used: signature-based and anomaly-based. Signature based IDS have signatures that detect attacks, but if new types of attacks come along or attacks with different approaches with new signatures that are not updated, then the system won't detect the attack. Whereas the anomaly-based IDS uses a machine learning approach in which the model is trained on previous attacks in order to predict attacks based on network traffic. Layered approaches are also used for the batter performance because the training of models and predictions takes heavy system resources as compared to signature matching.

The world today is threatened by various types of cyber attacks that can halt systems or restrict their functionality. The most common cyber attacks are DoS(Denial of Service), DDoS(Distributed Denial of Service), MitM(Main in the Middle Attack), SQL Injection, XXS(Cross Site Scripting), Malware attacks, APT(Advance Persistent Threat) etc. Each of these attacks has different subtypes. There are some past examples in which Cyber Attack become cause of heavy loss. Stuxnet is one of them which is the cyber attack from America on Iran which damage the Iran nuclear program. Recent example is Russia and Ukraine in which Russian hits the Banking sector of Ukraine. Blackout attacks and stealing of solders mobile location to hit the bases are also accountable.

Our topic is based on the combination of IoT, Edge Computing, and Cyber Security in which we develop a solution named IDS which is able to keep an eye on systems and detect cyber attacks. The topic aims to protect Edge computing from different cyber attacks and continuous monitoring of digital assets. We continually monitor the sensors and connected systems to the network are online or offline, in case of a problem system generates an alarm for the specific intrusion. For the network based cyber attacks IDS predicts the attack on the bases of network traffic and generates an alarm in case of network intrusion or cyber attack. We proposed an IDS which has two parts NIDS and SIDS. SIDS checks the sensor connection and the system state, as well as inspects the sensor data. In case of violation SIDS informs about the sensor and violation. As NIDS uses the machine learning techniques so it is anomaly-based, but we can still use the signature mechanism with it. So, for the next time if attack comes with the same signature, it can be detected by signatures instead of machine learning because ML requires more resources. Our IDS is also able to consume the other solution APIs to add the IoCs(Indicator of Compromises) for mitigation of attack. Like if the administrator uses the firewall as well with our IDS than he is able to connect the firewall with it so IDS also updates the IoCs as well in the firewall to mitigate the attack. This is how our IDS provides the connected threat defense. As a first step in building NIDS, we identify the benchmark dataset which provides a wide range of modern cyber attacks, then we perform some prerequisites on the dataset, and then apply the 6 Random ML algorithms for creating the model. After analyzing the different datasets, we choose one. Once the dataset has been selected, we will find the best algorithm that provides high accuracy within a short period of time. The selected algorithms are Random Forest, Decision Tree, Multi-Layer Perceptron, Logistic Regression, Naïve Bays and Support Vector Machine.

Cyber Threats are becoming more complex and increasing daily, so we need to develop protection tools that can detect modern attacks. So proposed IDS is installed on the edge server so it can detect the local cyber attacks and automatically add the rules for FW or ACL. The administrator gets visibility of the network due to the solution so he can see what's actually happening on the network. For critical industries like chemical or medical where IoTs are used vastly, we need to protect those for the safety of the environment and human life. Our solution is built for the generic attack type so it can be used in different IoT fields.

1.3 Research Reason

IoT is the most crucial technology of the 21st century and evolves daily. IoT has many applications in daily life like kitchen appliances, cars, smart homes etc. Different infrastructures use IoTs to improve performance, quantity and quality. The 21st century is a time of digitization where everything can be controlled using the computer system. The major infrastructures which use the IoT technology are Information Technology, Water and wastewater System, Energy, Nuclear systems, Chemical Industry, Food and Agriculture, Defense Industrial Base, Healthcare, Transportation, Financial Services, Dams, communications, Emergency Services and Manufacturing Industry.

Because some critical organizations use the IoT, they are threatened with cyber-attacks. As we already talked about the 5th Generation of warfare which is based on cyber-attacks. For defending those critical organizations' IoT infrastructure, we need defensive tools which able to detect the attack and mitigate the attack if possible otherwise, system administrators take the requisite decisions. Solutions are already developed and improve on daily bases because new type of attacks are introduced daily. So defensive tools also need to update and enhanced in a tactic to detect modern attacks. The system is built in old age unable to detect modern attacks, so we need to continues update in the system. We are developing the IDS which also has the capacity to keep an eye on the system and also detect modern attacks in a short time, then provide the connected defense for mitigation of attacks over the edge computing. The solution is useable for organizations those unable to afford other cyber security solutions so we build the general one with improved accuracy. Usually, IDS do not inform about the state of systems or sensors but our IDS is able to let about the state of sensors or systems and inspect data sent by the sensor to find the intrusion. If the sensor sends invalid data then IDS report it using the SIDS component, so if there is a sensor problem in the network, the system administrator gets the alert for intrusion, and then he will fix it on time.

1.4 Proposed Methodology Strength/Weakness

We built our ML model using a modern dataset containing samples of ten different modern attacks.

- If the attack comes with the updated signature, our IDS can detect it.
- In order to detect cyber-attacks, our IDS is designed to work on edge computing.

- Our IDS can keep track of both online and offline sensors and systems with intrusions, while also keeping an eye on the online and offline sensors and systems.
- Our IDS is not so much resource intensive.Because we use signature-based techniques as well so before applying ML we check the signature library. if signature is found we just alert insted of applying ML.
- Our IDS unable to detect the attack if attack comes with the new type so for that we need to update the ML model of IDS.
- If IDS not plug on core switch, then IDS unable to monitor the all incoming and outgoing traffic.

1.5 Problem Statement

Because IoT devices are resource constrained, they are susceptible to cyberattacks that affect their usability. Due to the latency in response, real-time decisions cannot be made in classical architecture because IoT devices send data directly to the cloud, which increases network bandwidth utilization and cloud load. Edge computing solves this problem while still being vulnerable to security threats. In order to protect IoT base systems from different cyber threats, we need to enhance the security of edge computing. As a result, we need an updated IDS system capable of detecting modern cyber attacks, as well as monitoring the sensors/systems and inspecting the sensor data to detect intrusions and useable on the edge computing. With its API consumption feature, the IDS is also capable of updating IoC in other security solutions, such as firewalls. So it can provide the connected threat defense.

1.6 Problem Questions

- Identification of Datasets that are highly suitable for development of IDS over Edge Computing?

- Are there any possibilities to integrate an independent edge computing system to detect intrusions from both ends?
- What can we do to make Edge Devices smart enough so that they can comfortably detect attacks on the network side and communicate with the cloud in a secure manner? (Anomaly Based)?
- The identification of a way to keep an eye on the sensor/system connected to the network, as well as the inspection of its data to detect intrusions into the network?

1.7 Previous Research Weaknesses

According to the previous research, cyber attacks growing on a daily basis and new types of attacks are being discovered which harm the digital infrastructure on a regular basis. As the IoT has vast use in digital world which rise new security problem that need to be addressed in order to ensure the safe operation of the system. In order to protect ourselves, we need cyber security defensive tools that help us to timely detect cyber attacks and give us a true picture of what is happening in the digital world otherwise, we can not protect what we cannot see. It is necessary to have an IDS that can detect modern network intrusions as well as inspect the data collected from IoT sensors to identify devices causing internal intrusions. For attack mitigation, IDS must be able to integrate with other security solutions, such as firewalls, to share IoCs and mitigate security threats. Also, we identify that most researchers protect fog nodes and clouds from cyber attacks but they do not focus on the end IoT devices, so IDS needs to be able to run on edge servers to protect end IoT devices. In addition to dataset selection and machine learning algorithms, they play an important role in detecting attacks. IDS use machine learning so also become anomaly based IDS with signature base. Fund research weakness in [2] is also being addressed by achieving the highest possible accuracy without a layered approach to stay away from the resource intensive techniques.

1.8 Objective

The objective of this research is to design an IDS which is suitable for Edge Computing to protect the IoT infrastructure from today's modern cyber-attacks and keep visibility of the system and sensors. In addition to inspecting data sent by sensors, IDS also analyzes the intrusions they generate. In the event that the sensor generates false data, the system alerts the administrator so the sensor can be checked to prevent damage. The purpose of an IDS is to protect the network from both external and internal threats. The IDS also provides connected threat defense by consuming APIs from other solutions to update IoCs in other solutions, such as firewalls or ACLs, to prevent attacks. Security field required the continues update in defense solution otherwise old age solutions unable to detect the modern attacks so we need the complete research in this field to identify the new security threats and then update security solutions after certain time to ensure defense.

1.9 Motivation

Cyber attacks are increasing daily, and new types of attacks are being discovered on a daily basis because of security threats.

We cannot stop using this modern technology which automates our daily tasks and makes our lives easier. To overcome security threats and to use this technology in a safe manner, we need to make and update security solutions. Every day, researchers identify attackers' new approaches and then highlight defensive measures to prevent them. As a result of the research, enterprise companies develop new tools or update existing tools to ensure their security. We also have different datasets available to build the IDS which can be utilized on edge computing. Machine Learning algorithms and techniques also get mature so we apply the ML on datasets to use it in the best manner to build the security tool. In order to use modern technology safely, we need a strong defence, otherwise, attacks might gain access to it and misuse it. For keeping track of the network, we already have some network sniffers which can be used. As a result, building an IDS for edge computing is possible since we have the tools and technologies to use.

1.10 Thesis Organization

The whole document is divided into five different chapters. In chapter one there is a background knowledge for the domain with the brief introduction about the topic, reason behind the research along with problem statement, research questions, objective and motivation for the topic. The literature review is presented in chapter two. After conducting a literature review, we identified the techniques previously used for building an IDS. We also discuss the research gap that was identified, followed by our expected approach, a brief methodology, and a conclusion. In chapter three we present the details about the proposed research methodology along with the deployment architecture. Deployment architecture diagrams are also being shown. In chapter four, research discuss the detailed about the experiments and obtain results. Chapter five discuss the conclusion and future work.

Chapter 2

Literature Review

2.1 Introduction

In order to identify the gaps in research and the current tools and technologies available for protecting IoT infrastructure, we conducted a comprehensive literature review that identified the fields in which IoT is used and what tools and technologies have been used in the respective industry. It is also important to discuss the security threats associated with respective industries. Next, we discuss the tools that can be used to protect IoT infrastructure and architecture. Afterwards, we discuss the use of Machine Learning in security and tools used for IoT-based security. Our next discussion will be about the dataset that we used to train the machine learning algorithm. Algorithms have much importance to achieve the maximum level of accuracy, so there is a need to discuss them here as well. In the end, we highlight some research gaps and limitations.

2.2 Existing Methodologies

The term Internet of Things came into being in 1999 when a scientist kevin Ashton proposed the use of radio frequency identification chips in products to track them. Then IoT industry get started and evolved on daily bases. S.Balaji and karan

Nathani in [3] conduct the survey in 2019, they identify the industries whose use the IoT with respect to time. According to authors there is a rapid change in industry which use the IoT in development of IoT technology itself. There are huge advantages of use of IoTs but still some disadvantages are there which author discussed. According to the survey security industry use the IoT with the highest rate from 2013 to 2018. Then protocols are at the second stage, Healthcare is on number three, smart city at the four, and agriculture is at umber five. Because security industry use the IoT maximum examples are IP Camera, RFID based ACMS etc. so author also discuss the security challenges because if IoT itself is not secure than what if someone compromise the IoT security than defiantly he compromises the security of system for which the IoT is used. In this survey author just highlights the major points instead of deep discussions.

In 2019 a survey is conducted IoT on security, application areas its security threats and solution architectures by Vikas Hassija and others in [4]. According to research IoT improve the level of comfort in humans' life but still needs the high level of security, privacy and mechanisms to recover from the attacks. This study presents a thorough analysis of the problems and potential security flaws in the IoT Technology. After covering the security concerns, we'll go into the different new and current solutions that aim to instill a deep sense of trust in IoT Technology. Research discus the four major and different technologies to overcame the security issues of IoT which are Block Chain, Fog Computing, Edge Computing, Machine Learning. According to the paper authors present the previous architecture used in IoT in which IoT device directly establish the connection with the cloud. Then the current architecture in which fog or edge node is present before communication with the cloud to enhance the security challenges. The future IoT architecture of IoT is end to end based in which each IoT device directly communicate with the requisite destination instead of fog, edge or cloud. Discussed Application areas are smart cities, smart environment, smart metering and grids, security and emergencies, smart retails, smart agriculture and animal farming, and home automation. Then Author discuss the security issues at each layer of IoT including sensing, network, middleware, gateway and application layer along with the possible cyber attacks on each layer. Author discuss in detail blockchain, fog computing and edge computing use in IoT and how these technologies enhance the IoT security. From paper we get an idea to use the edge computing for IoT devices and then develop or enhance security tool to deals with the security issue. Author just present how's the edge computing overcame the security threat but did not discuss the security solution for protection in case of cyber-attack. Because in secure architecture cyber attacks are still possible and we need a system to detect those attacks to prevent from the furthers damage.

In 2021 Fazlullah Khan and other members of IEEE provide the research article [5] on secure and intelligent communication scheme used for IIoT based edge computing. According to the research IoT devices use in Industry are mission critical and needs the immediate response to operate the Cyber Physical Systems, for response real time data processing is also needed but due to latency in cloud delay occurs so there is a use of edge computing in Industry. Edge computing helps to achieve the real time data processing so immediate response is possible. According to the author pervasive edge still face some challenges in terms of secure communication, network connectivity and resource utilization on edge server. To address these issues, they proposed secure and intelligent communication scheme in which forged identities like Sybil are detected by IIOT devices and share with edge servers to provide upstream transmission to their malicious data. If sybil attack is detected than in response each edge server start execution of pABC to form best possible network configuration. Each server start the job migration to balance the job loads. Author proof his concept using the experiments, but still other cyber attacks are possible. These Cyber attacks might harm the entire system so still we need the proper solution to detect the other possible cyber-attacks and prevent them to cause more damage. So the reason we are proposing the IDS for the edge computing.

In 2018 another research from IEEE is also conducted in [6] which provide the survey for 26 Trust Management Techniques. Trust management is very much important in IoT to prevent from some cyber attacks those carried for information stealing of misuse of system. Author conduct the comprehensive study on trust management systems and present the short summary for each in which he define advantages and disadvantage of technique. This is also useable in the Edge computing and will be part of out future work because currently our focus is to develop and IDS which is able to keep and eye on entire network and detect the cyber attacks.

Teklay Gebremichel and other members discuss IIoT security, privacy, and their current standards and challenges in [7]. In the paper author discuss the features and enhancement of security and privacy breaches on sensitive information. The complex nature of IIoT reveal some security problems. For the safe and reliable operations, we need the proper security controls. The main idea behind the paper is industrial internet of things where he discusses the standards and future challenges on IIoT. Author also claims that IIoT are vulnerable to different security threats on communication and connectivity. In current security of IoT research article discuss the IIoT edge connectivity protocols including Bluetooth, Zigbee, IEEE 802.15, NB-IoT, WirelessHART, LoRaWAN, ISA 100.11a and 6LoWPAN. For platform connectivity different protocols also discussed name as CoAP and MQTT. Article focus in on basic protocols for achieving the security and authentication, access control, identity management, key management, data isolation etc to achieve the security. But our focus is on the network layer to detect the cyber attacks where the paper focus is on discussing the security on basic protocols. These protocols are black box for us.

In 2020 Mohammed Atiquzzaman with other author presents and research article [8] in which they introduce the framework PriModChain. Article basically highlight the issue of use ML in IIoT because their ML models are trained on sensitive data but ML leak privacy of data in case of adversarial attacks. So author framework enforce on privacy and trustworthiness. Author Perform simulation using the python-based network programing and proof his concept.

Resul Das and Muhammed Zerkeriya presents the research article [9] in 2019 which covers the Analysis of cyber attacks on IoT based Critical infrastructures. The article discusses the different types of cyber attacks and precautions for preventing them. The author also examines the critical infrastructure and common cyber attacks with approaches to mitigate them. Discussed Cyber attacks are Tram

Hacking, Power Company Hacking, Stuxnet, Water Distribution System Hacking, Power Grid Hacking, Dam Cyber attack etc. Using IoT infrastructure, the author provides a complete list of attacks that demonstrate the importance of cyber security. Several security measures are recommended by the author in order to mitigate cyber-attacks, such as: Access Control System, Encryption over data, Authentication, Physical Security, Backdoor and Login Process, and Intrusion Detection System. It is important to consider the author's suggestions because preventive measures alone are not enough; we need a solution that continuously monitors systems and detects intrusions so that we can mitigate the effects of a cyber attack.

In [10] Author discuss the authentication schemes for security in combine edge, fog and cloud computing. Author proposed the light weight authentication scheme after finding some security flaws in the fog computing. Once the author define authentication is done both fog and participant agree on a session key to encrypt the subsequent message. When there is encrypted communication between direct sensor and cloud than we unable to check whatever sensor is sending. So for keep an eye on the network we still need to use the certificates on IDS to decrypt the sensor data and analyze the intrusion. Yes, if we use the authentication system than there is an extra layer of security but we could not detect if sensor get hang or damage and sending the improper data. Example of this is that if there is a too hot weather and sensor send the data in negative so there must be intrusion.

In [11] author discuss the SDN based edge computing for healthcare system. SDN is software define network in which traffic network traffic purely software based. SDN Play an important role to enhance the security but most of the technology still not adopt the SDN due different reasons. In the research article they design secure framework in which devices are authenticated from edge server with light weight scheme then sends data to edge for storage and processing. There is a SDN controller which also balance the load over the edge servers. Author also evaluate his framework using computer simulations. The concept is also useable in our future work but author miss the IDS or other security tool which is able to detect the intrusion on SDN network so there is still need of IDS.

Thrung Thu Huong with Other member in article [12] presented in 2020 proposed the attack detection mechanism on edge computing because it is near to end system for taking the requisite actions. Author named the mechanism as LocKedge (Low Complexity Cyber attack Detection in IoT Edge Computing). Author perform his experiments our the most updated BoT-IoT dataset. Author use the NN, CNN, RNN, KNN, SVM, RF and Decision Tree and compare those with the define algorithms. At the end of his research author proof that his define LocKedge works batter in terms of performance. LocKedge has less training time but high accuracy,.Author Also monitor the network traffic using the Raspberry pi3 and monitor the traffic but when he apply the sample of 400 to 2400 per second CPU and RAM utilization get too much high. According to author if there is Pi4 than System should perform batter because of its resources are more powerful.

In article [13] Authors present the user behavior base anomaly detection system for smart homes. This method represents user behavior as sequences of user events, including Internet of Things (IoT) device functioning and other observed activities. Authors technique learns event sequences for each home state, such as time and temperature, based on how users behave under those conditions. To limit the impact of other users' events in the monitored sequence, authors method generates various event sequences by eliminating certain events and learning the most commonly seen sequences. For the sake of evaluation author conduct the experiment in which he shows that authors model detects the 90% of anomalies. In article author compare his model with the Hidden Markov Models. Authors system able to detect the anomalies which are based on user actions but in cyber security attackers perform the digital attacks. So this solution is not enough to ensure the proper digital security.

In [14] K.A Sadiq, A.F Thompson and O.A Ayeni provides a conceptual framework which can be utilized for the mitigation of DDoS attack in cloud Network using the fog and SDN. Here fog node act as a additional firewall to mitigate the attack. Author proposed technique is that first of all packet comes and system checks it TCP header in case of spoofed packet system simply drop the packet, otherwise packet pass to flow table in that case packet could not send to SDN controller which inspect the packet using classifier algorithm in case of anomaly SDN drop the packet but in case of success packet also pass to threshold check if there is threshold system drop the packet otherwise packet reach the destination. This research is focus on mitigation of DDoS attack other attacks are still possible here. Fog node is also away from the actual cloud so attack from the inside is still possible and system will not detect due to limitation of deployment architecture. The reason we select the edge computing for attack detection because edge servers are near to actual infrastructure.

Article [15] presents the use of IoT, in medical for applications which provide the remote care and diagnosis facility. According to the author because health data is multi dimensional and machine learning promise and provide the best possible solution for intrusion detections. Mostly authors use their personalized data or network flow to detect the intrusion but here author use the both at the same time for attack detection and gather the batter efficiency. Author build the real time EHMS (Enhanced Healthcare Monitoring System) which take the both feature personalized data and network flow. The security threat is possible in the architecture because author sends the data to cloud where Man in the Middle Attack is still possible. System applies the different ML algorithm to detect the attack. Author results shows that author performance is increased by 7% to 25%. Here data is send to cloud for the processing and detection of attack which is not a good because in case of some cyber attack system gets isolated from the cloud so there is need to apply the solution which is actually near to IoT infrastructure for detection of attacks. We suggest to use the training, testing and real-time detection on edge layer. The author focuses on threats related to data alteration and spoofing. In this paper, the key contribution is the design of a healthcare testbed that can be used for future research, the collection and analysis of new healthcare datasets that combine the two features, and a security system that does not burden sensors with attack detection, and using a different algorithm to detect attacks in real time. The used algorithms are RF (Radio Frequency), KNN (K Nearest Neighbor), SVM (Support Vector Machine) and ANN (Artificial Neural Networks).

In 2018 research article [16] presented by nadia Chaabouni and other discusses the security threats over the IoT and importance of NIDS. Author survey and classify the security threats for IoT networks by evaluating the existing defense techniques. Research article mainly focuses on the NIDS so article review the existing NIDS implementation tools and datasets as well as free and opensource network sniffing softwares in context of IoT architecture, detection methodologies, validation strengths, treated threats and deployment algorithms. Author use the Machine Learning for improved results in NIDS. Author provide the comprehensive research review over NIDS deploying different aspect of learning techniques for IoT. Authors say's on the network each IoT device has a unique IP address which is used for communication over the network and traditional NIDS works on rules and signatures which leads to high false positive, false negative in attack detection and these systems don't have capability to detect the new attacks because of unavailability of signatures. Researches proposed the use of AI (Artificial Intelligence) and ML with deep learning algorithms for improvement in such NIDS to detect the new type of attacks. So, paper surveys and evaluate different ML contribution to use in NIDS.In Figure 2.1 author present the different Threats which are classify in different challenges.

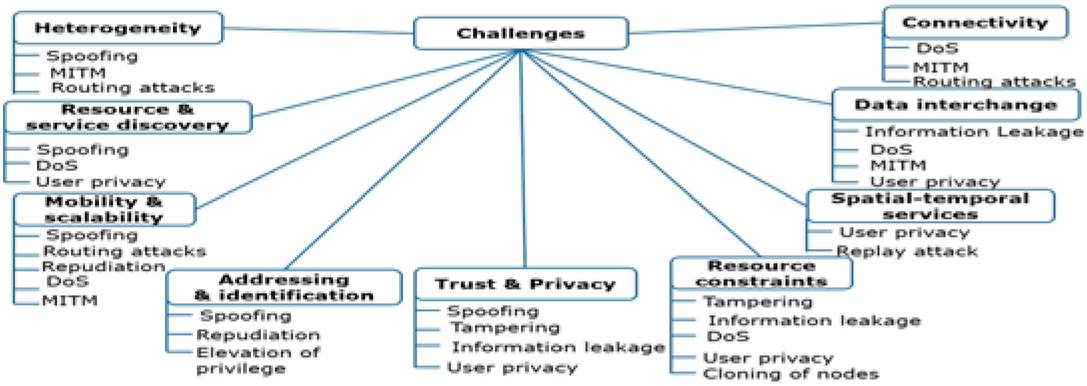


Figure 2.1: Presented Challenges in article [16]

Author discuss the 6 traditional threat defense mechanisms Filter Packets, Adopt Encryption, Password Authentications, Audit of logs, IDS and IPS. Author build the NIDS on 4 steps. 1 collect the traffic data from the network, 2 Analyze the collected data, 3 Identify relevant security events and 5 detect and report malicious events. Author also provides the Comparison on Datasets the used datasets are KDD99, NSL-KDD, UNSW-NB15, Sivanthanet al. Dataset, CICIDS and CSE-CIC-IDS2018. Than Author Provides the comparison for Open Source network sniffers named as Tcpdump, Wireshark, Ettercap, Argus and EtherApe. Discussed Open-Sources NIDS are snort, suricata, Bro-IDS, Kismet, OpenWIS, Onion Security and sagan. Author also present the different ML Algorithms shown in below figure.

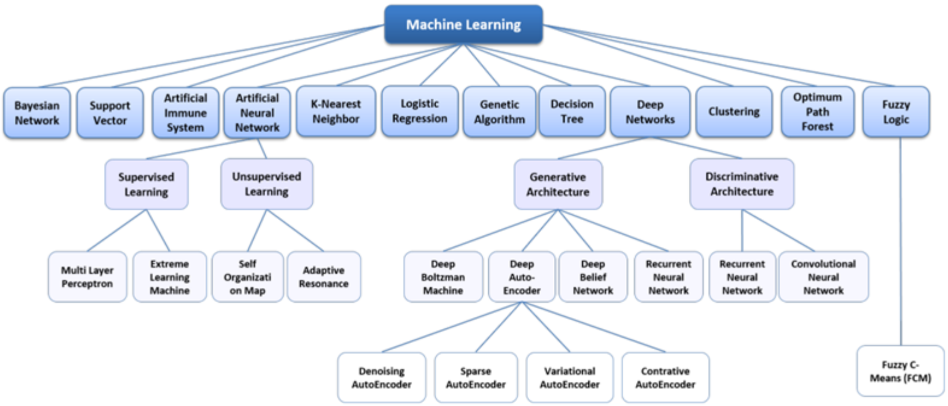
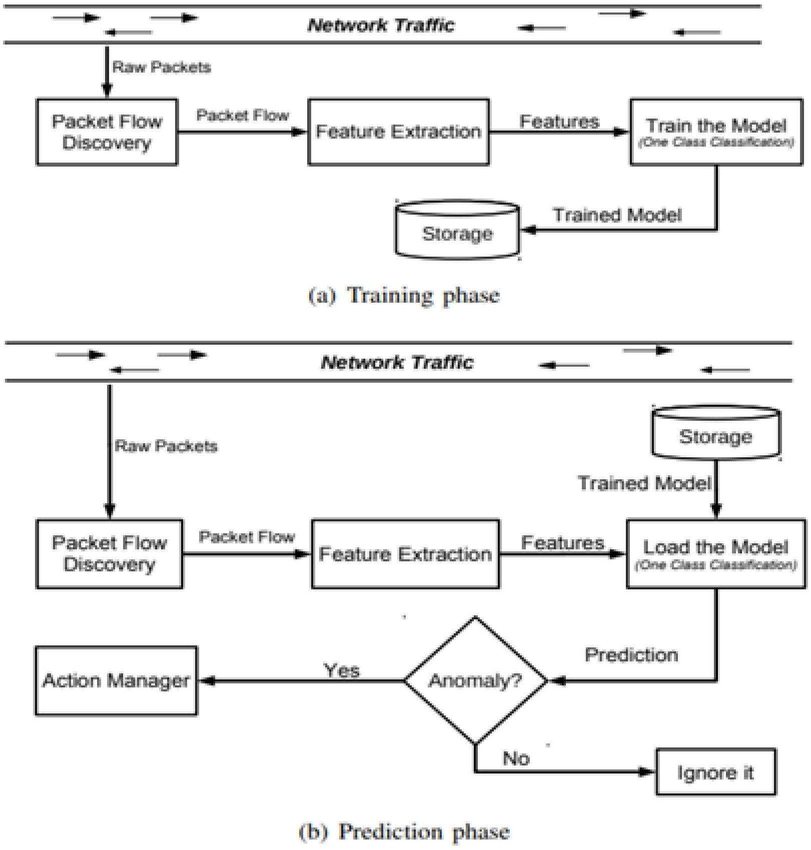


Figure 2.2: Shown ML Algorithms in [16]

In 2019 a research article [17] presented by Fariz Andri Bakhtiar with others. In the article author use the ML J48 algorithms for overcome the DoS attack. Due to DoS attack data produced by the sensor not reach to end destination so system fail to works. Author claims 100% detection for intrusion while his system capturing the 75% of network trafic. Author collects the dataset and processed by Weka tool which have J48 as implementation of C4.5 algorithm. Then the Processed Data generates the decision tree which used as decision engine. Very firstly author clean the data and use the selected features which gives the best possible performance. Author major focus is over the detection of DoS, other attacks are still possible so there is a need of IDS which able to detect the other intrusions as well.

An article [18] by Mojtaba Eskandari discusses Passban IDS in 2022. The Passban Intelligent Intrusion Detection system detects intrusions on each device directly connected to it. It is best to deploy the proposed solution over cheap gateways, such as single-board computers, in order to detect attacks near data sources. The Passban program is capable of detecting port scanning, HTTP and SSH brute force attacks, and Syn Flood attacks with very low false positive rates. Ultimately, the paper aims to make a lightweight Anomaly IDS that is suitable for IoT gateways. The Passban has 2 phases one is the training and the other is the prediction phase. It also composes of packet flow discovery block, feature extraction block, Train model block, Action manager procedure and web management interface. Passban Architecture is shown in below.



(a) Training phase

(b) Prediction phase

Figure 2.3: Proposed Passban Architecture in [18]

The Passban algorithm learns the normal behavior of the system using machine learning algorithms. After the training phase, the model is trained to detect anomalous network traffic events. The author uses supervised learning to distinguish between malicious and legitimate traffic. As the model is trained over the training data with classical attacks, and the model classifies the results into defined classes when a new class occurs, the system fails to respond, so the supervisory learning approach is more useful. In addition, the author discusses the challenging situation in which new attacks occur. The author tests his proposed IDS under various attacks and demonstrates its capabilities. Using two methods, the author deployed the Passban, one over the gateway and the other over a separate device. The Passban itself gets trained over the normal flow of traffic. The Passban detects the attack successfully, but it's not as accurate as it should have been.

Article [19] discuss the instruction detection system based on machine learning to detect the network attacks on IoT in 2022. The primary focus of author research is to apply the supervised machine learning algorithm to build the IDS. Author use the UNSW-NB 15 dataset than apply the different pre requisites operations on it than author apply six selected algorithms over it name as PCA-XGBoost, PCA-CatBoost, PCS-KNN, PCA-SVM, PCS-QDA and PCA-NB. Author 2 algorithms achieve the accuracy level of 99.99% can can be utilize on Smart home, smart city and health care IoT infrastructures. Authors IDS is best in terms of accuracy but author IDS is unable to keep eye on the system in which IDS able to check the states of the sensors and inspection of data sanded by the sensors.

In article [1] Yinhao with other members provide the detail survey on influential and basic attacks with the corresponding defense mechanism for edge computing. The primary focus of the paper is to discuss the possible cyber attacks on the edge server or edge device. Discussion of attacks are group into major type and than author talk about the maximum possible sub types of attacks and also suggest the countermeasures to avoid from the attack. In the his research article author not going to suggest the security tool which help the administrator to detect and than prevent from the attack.

The article [20] presented by Mossa Ghurab with other members in 2021. In article they provide the complete analysis of different bench mark datasets which use be use to build the IDS form mitigation and detection of modern cyber attacks. The covered datasets are KDD99, NSL-KDD, KYOTO 2006+, ISCX2012, UNSNW-NB 15, CISSD-001, CICIDS2017, and CSE-CIC-IDS2018. Author examine the each dataset and provide the detail anylsis for instances, features, classes and nature of features. So this information helps to select the dataset before designing the NIDS for specific system. Author recommend the CICIDS-2018 but it covers the 7 different classes. According to our requirement we found the UNSW-NB 15 as a good dataset but we need the more research on it before choosing it. Dataset is an essential part for building the machine learning model so if we not select the best possible than we unable to achieve the required security.

In article [21] Nour Moustafa and Jill Slay provide the statistical analysis of UNSW-NB 15 and KDD99. Author discuss the lacks in KDD99 for the reason researchers develop the UNSW-NB 15. In this study, authors demonstrate the UNSW-NB15 data set's complexity in three aspects. First, an explanation of the statistical analysis of the observations and attributes is provided. Second, there is an evaluation of feature correlations. In third and last author apply the five existing classifiers which evaluate the complexity in terms of accuracy and then author comaapare thr results with KDD99. Author proof KDD99 as less complex dataset where as UNSW-NB 15 is more complex in nature because it deals with the modern attacks. The dataset can be used in NIDS to evaluate it with existing once.

For the more details about the UNSW-NB 15 we also study the article [22] presented by same authors Nour Moustafa and Jill Slay. Authors provide the suggestion for comprehensive dataset to NIDS. Author provide the comparison for the KDD98, KDDCUP99, and NSLKDD. Different studies shows that these dataset are now old age and there machine learning models are unable to detect the modern attacks, so author also examine the UNSW-NB 15. After that author discuss the each dataset in details like how the dataset was build and which cleaning methods use to extract the right data from the cap files. After the complete research and comparison author in the end proof the UNSW-NB 15 is the bench mark dataset which have the maximum number of attack classes and its is a modern dataset so its machine learning model is able to detect the new modern attacks.

In article [2] Souhail Meftah with others made the NIDS using the UNSW-NB 15 dataset. The very firstly they perform the Recursive Feature Elimination and Random Forest to select the best feature for use in machine learning. Than they perform the binary classification in order to detect the intrusive traffic from the normal network traffic. Used algorithms are Logistic Regression, Gradient Boost Machine, and Support Vector Machine. SVM gives the highest accuracy of 82.11%. After that author feed the SVM output to multinomial classifiers for improvement of accuracy. Author also evaluate the performance of Decision Tree Naïve Bays and SVM. Decision Tree gives the highest accuracy and F1 Score and layered classification improves the accuracy by 12%. Layered classification is shown in figure below. Due to these we only able to detect is there intrusion or not but we are unable to find the intrusion class like which type of attack is on going under the network. So this information helps the administrator to keep an eye over the network.

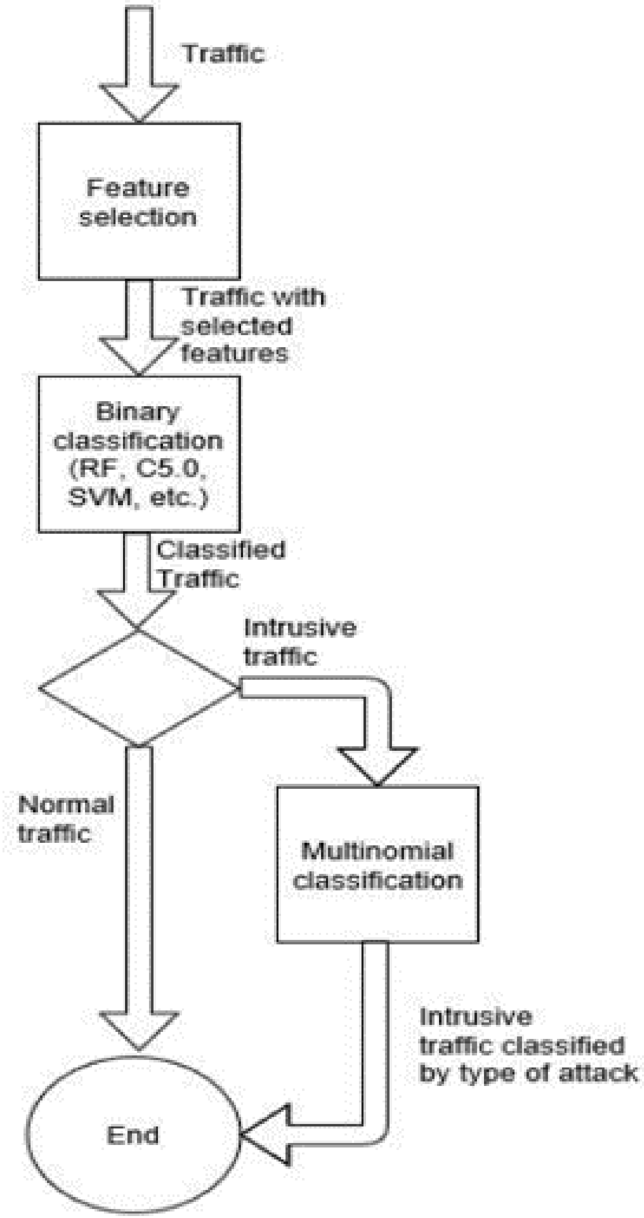

Traffic
Feature selection
Traffic with selected features
Binary classification (RF, C5.0, SVM, etc.)
Classified Traffic
Intrusive traffic
Normal traffic
Multinomial classification
Intrusive traffic classified by type of attack
End

Figure 2.4: Presented Research Methodology [2]

2.3 Strength/Weaknesses of Previous Research

Strength and weaknesses of existing research work are discussed below the in the form of table.

Table 2.1: Strength/Weaknesses of Previous Research

No	Article	Strength	Weakness
2019	[3]	• Discussion about the boom of IoT technology and its use in the different industries and associated problems..	• Author just highlight the issues and solutions are not being proposed.
2019	[4]	• Presents the major flows in IoT security. • Discussion on 4 Different security architectures which help to ensure security. • Discussion of security threats on each layer.	• Did not discuss the security solution in case of cyber attack. • No attack detection mechanism is being proposed.
2020	[5]	• Arises the requirement issue of real time response in IoT and highlight the solution. • Proposed the solution for Sybil Attack	• Other Cyber Attacks are still possible.
2018	[6]	• Highlight the importance of trust management.	• Just provide the survey and discussion don't focus on implementation of best technique to be used in edge computing.
2020	[7]	• Article focus on the security of basic protocols and technologies use for communication in IIoT.	• Author completely ignore the cyber attacks possible on system architecture.

2020	[8]	• Article highlight the use of ML in IIoT which cause the extraction of sensitive data. • Proposed the framework for privacy and provide its simulation.	• Author ignore the cyber attack possible on using ML models in IIoT and ignore the threat on use of system without security solution.
2020	[9]	• Covers the analysis and precautions of cyber-attack possible on critical infrastructures use IoT.	• Author ignores the threat detection solutions because precautions are just basic steps but there is still need of tools those detect the attack.
2021	[10]	• Author proposed the light weight authentication scheme after finding some security flaws in the fog computing.	• Author proposed solution is good in authentication but this system do not have other features to detect the state of sensor and possible cyber attacks.
2020	[11]	• Author proposed the use of SDN for IoT security. Author proposed the framework which authenticate the sensor with edge computer for reliable connection.	• SDN is new concept mostly industry do not use it still so we need of solutions which can also be used with previous and current edge computing architectures.
2021	[12]	• Author proposed the Low Complexity multi attack detection LocKedge security solution which provide the security on edge computing and also deploy able on small chips. This helps to achive the detection machanisam near the source so we can protect it.	• It use the BoT IoT dataset which covers the low number of attack classes and supported protocols.

2020	[13]	• Author proposed the behaviors based anomaly detection for smart homes which relay on human activities.	• This cover the Physical security of home but when there is cyber attack there is no visibility of sequence of actions because home come under the digital attack so this solution is not enough to detect the cyber attacks as well.
2020	[14]	• Autor Provides the conceptual framework for mitigation of DDoS attack in fog and SDN.	• Research focus on the detection of DDoS attack on fog node where as actual sensors are away from the fog node those are under the threat of cyber attacks and there is not solution to detect and defense those attacks.
2020	[15]	Author build the real time EHMS (Enhanced Healthcare Monitoring System) which take the both feature personalized data and network flow and detect the attack more accurately. • Author rise the performance of attack detection by 7% to 25% using his concept. To build proper and realistic intrusion analysis, a new dataset was collected and analyzed that combined network flow and biometrics information.	• Author machine learning models are hosted on cloud so in case of dysconnectivity system fails to detects the attacks. • Used algorithms are consumes too much system resources.

2019	[16]	• Author survey and classify the security threats for IoT networks by evaluating the existing defense techniques. • Focus on building IDS using open source tools and technologies • Researches proposed the use of AI (Artificial Intelligence) and ML with deep learning algorithms for improvement in such NIDS to detect the new type of attacks. • Autor also discuss the different old and modern datasets and machine learning algorithms to use in building of NIDS	• Author simply focus on the attacks and modern IDS which is able to detect modern attacks but author prosed IDS is not able to keep an eye over the system.
2019	[17]	• Author use the J48 light weight machine learning algorithm to detect DDoS attack. J48 is also knows as C4.5 which is a type of Decision Tree Algorithm. • Provide the complete simulation for detection of attack and guaranty the accuracy. Because author extract the gain ratio for all features and select the best one.	• Other attacks are still possible which could cause the service interruption.

2020	[18]	• Author proposed the Passban light weight Intelligent IDS which able to detect the intrusion on device connected to it. • This covers the multiple attack classes and implementable on IoT gateways to detect the intrusions.	• Accuracy of Passban is not much good. • Covers the classical attacks.
2022	[19]	• Author proposed the IDS which use the modern dataset and covers maximum protocols and attack classes. • Author achieve accuracy level of 99.99%.	• Author used Algorithms are not light weight. • Authors proposed IDS is not able to detect intrusions produced by the sensors.
2019	[1]	• Provide the detail survey of cyber attacks possible on edge computing architecture. • Author coves the maximum possible attack type and their sub type.	• Author do not suggest the security tools which can be utilized to mitigate the security threat. • Author do not provide the implementation of his research work which guaranty the author claims.
2021	[20]	• Author provides the detail analysis and comparison of recent bench mark datasets which can be utilized to build anomaly based IDS. Author covers the nature, classes and features set of the dataset in comparison	• Author do not provide the accuracy of datasets models in real environments.

2016	[21]	• Author provide the statistical analysis of UNSW-NB 15 and KDD99. • Also implement the ML algorithms to evaluate the complexity level of each dataset.	• Author do not provide the accuracy of datasets models in real environments.
2015	[22]	• Author also provide the suggestion of selection of dataset for anomaly based NIDS. Author provide the complete study and discussion on each dataset.	• Author do not provide the accuracy of datasets models in real environments.
2019	[2]	• Author Build the NIDS using the feature Layered approach to detect the modern attacks. • Author achieve the accuracy level o f 86% after applying the different machine learning algorithms.	• Layered Approach consumes two much resources because data is based twice to predict the state as attack or normal.

2.4 Discussion on Previous Research

Various researchers focus on different areas of IoT security, some only focus on cyber security, and others suggest practical tools that can be utilized to defend against cyber-attacks. While some researchers improve the performance of security tools by applying different techniques, other creates the datasets that are the basis for building machine learning-based security tools. Our research work begins with a literature review, which we conduct as a baseline. In [3] author highlight the details Industries those use the IoT and discuss the security challenges they face due to use of that technology. These industries are highlighted as a result of this research. Authors discuss IoT based security issues in [4] and suggest ways to mitigate those threats with different technologies. This research provided us with different ways to implement the IoT in Industry in order to mitigate cyber attacks. A discussion of the mission-critical nature of Industrial Internet of Things can be found in [5]. In order to achieve real-time responses that cannot be achieved from the cloud directly, the author suggests use of edge computing concept. Since IoT infrastructure is based on distributed IoT devices, so trust is a fundamental requirement. Security controls should also be implemented on IoT devices and proper trust should be maintained said by the authors in [6] and [7]. According to the author [8], there are some issues regarding the use of machine learning in IoT because there are some attacks where attackers reveal sensitive information from the machine learning model. so we have to take care of those and try to mitigate the cyber attacks causing the data exfiltration. In [9] author discusses the different savvier cyber attacks possible on IoT based infrastructure. The author also discusses the use of authentication, authorization, and encryption in IoT infrastructure to overcome security issues. Similarly, in [10] the author proposed a lightweight authentication system for IoT use. To overcome cyber threats on IoT infrastructure and balance its load, the author suggests using SDN in [11]. This article [12] is important to discuss because it proposes the Edge Computing base IDS, which is very helpful in our research to conclude some parts of the idea. In this study, different machine learning algorithms were used on the BoT IoT dataset. In [13], the author also proposed the detection of anomalies in smart homes based on user activities but this concept cannot be used to implement the general IDS tool. In [14], the author suggests the use of TCP packet inspection on Fog Node and SDN for mitigation of DDoS attacks, but this only protects the Fog node from the attack, not the IoT devices that are the source of the data. As illustrated in [15], author designed an enhanced healthcare monitoring system that was able to detect some cyberattacks, which illustrates the importance of making the application as intelligent as possible so that it can detect attacks using a variety of techniques. In [16] author provides a comprehensive review of NIDS and also proposes NIDS using AI and ML for detecting attacks with new signatures. In [17] author also builds the IDS using J48 ML for detecting DoS base attacks. Also in [18] the author built a lightweight IDS called Passban. Different types of network attacks can be detected by Passban. As signature-based techniques are quite old, the author uses machine learning to predict new attacks. Even though the author's experiment with Passban was successful, there is some doubt regarding its accuracy level. With the different machine learning algorithms used by the author in [19], the accuracy level was improved to 99.99% using UNSW NB-15 dataset. Despite good detection abilities, still this IDS is unable to track internal systems and can't detect intrusions caused by sensor-based attacks. The IDS is effective for modern attacks, but is not able to detect intrusions from sensor-based attacks. Since the topic is already accurate, there is a small window for future research. [1] discussed the possible attacks on edge computing architecture but did not suggest ways to mitigate them. Different datasets with their characteristics are presented in [20], [21], and [22] to determine the optimal dataset to create a NIDS smart enough to detect modern attacks with the maximum possible number of classes. A layered approach is presented in [2] to build an NIDS, but the author achieves a level of accuracy of 86%. In this case, the author applies machine learning twice, which is a resource-intensive process. As the system is built on UNSW NB 15, the author in [19] achieves the highest level of accuracy using machine learning, so we can analyze this and improve it by adding some new features.

2.5 Research Gap

According to the literature review, cyber attacks are increasing on a daily basis, new types of attacks are being discovered and attacking the digital infrastructure on a regular basis. As the IoT is used more and more in the digital world, new security problems arise that need to be addressed in order to ensure the safe operation of the system. Since IoT has small memory and processing power, it is vulnerable to huge attacks. In order to protect ourselves, we need cyber security defensive tools that help us to timely detect cyber attacks and give us a true picture of what is happening in the digital world; otherwise, we can not protect what we cannot see.

IDS plays a key role in detecting intrusions on the network because most attackers attack the network side. Therefore, we should use IDS to detect intrusions in time, and then mitigate them with techniques. It is necessary to have an IDS that can detect modern network intrusions as well as inspect the data collected from IoT sensors to identify devices causing internal intrusions. For attack mitigation, IDS must be able to integrate with other security solutions, such as firewalls, to share IoCs and mitigate security threats. Also, we identify that most researchers protect fog nodes and clouds from cyber attacks but they do not focus on the end IoT devices, so IDS needs to be able to run on edge servers to protect end IoT devices.

In addition to dataset selection and machine learning algorithms, they play an important role in detecting attacks. IDS use machine learning so also become anomaly based IDS with signature base. Fund research gap in [2] is also being addressed by achieving the highest possible accuracy without a layered approach to stay away from the resource intensive techniques.

2.6 Expected Approach for our Research

Various approaches can be used to enhance IDS by different researchers. When problems are identified, we combine the knowledge and build the IDS that protects end users from a variety of cyber threats. During the development of IDS solution, we selected the best possible data set which covers a variety of attack classes and its machine learning model is also able to detect modern attacks. In order to protect end-user IoT devices against cyber attacks, we choose edge computing architecture. The IDS is broken down into two sub-parts: SIDS and NIDS. A sensor-based intrusion detection system is also being introduced. It inspects data from the sensors and detects intrusions. It also keeps an eye on the network and gives a clear picture of it. In order to provide a useful NIDS on edge computing, we use the best possible dataset and machine learning algorithm for higher accuracy without making the process resource intensive.

2.7 Conclusion

The literature review says that cyber attacks are getting worse every day, that new types of attacks are being found, and that the digital infrastructure is constantly being attacked. As the Internet of Things (IoT) is used more and more in the digital world, new security issues come up that need to be fixed to keep the system safe. So after founding some research gap we find the techniques to full fill those and produce the IDS according to expected approach.

Chapter 3

Proposed Research Methodolog

3.1 Introduction

We identified some research gaps in previous research during the critical analysis of the literature review, and for the sake of improvement we proposed the system with a complete methodology. For defence against new threats, cyber security needs to be constantly updated and improved. In our solution, the layered approach has been removed in lieu of using one model in order to improve the accuracy of the research work in the article [2]. In addition, our proposed system is capable of detecting online and offline sensors and systems as well as inspecting the data sent by the sensors to detect intrusions. The purpose of this chapter is to describe in detail how the proposed solution achieves an effective IDS that is suitable for edge computing and detects modern attacks through a complete methodology which is detailed in this chapter. A full discussion of the proposed system is presented in section 2, and discussion of the methodological approach is provided in section 3.

3.2 Proposed IDS

Cyber security has much importance in safe use of digital technologies which are connected over the network. For timely detection and proper mitigation of cyber attacks we have to use such security solutions. On the basis of research gaps, we found the in literature review we are proposing the IDS which have two sub modules to detect the network intrusions and sensors intrusion. SIDS will detect the sensors those are not responding properly and having problem in sending data while NIDS having signature and anomaly base engines to detect the network intrusions. Detail of each system will be discussed below. Protection of end IoT devices are much important so our proposed IDS is useable on Edge server for detection of attacks. Proposed IDS also have capability to consume the APIs of other solutions like firewall so it can update the IoC for the mitigation of attack known as connected threat defense. Components of proposed IDS are presented in figure below. We also discuss each component in detail where we highlight the importance and use of the component.

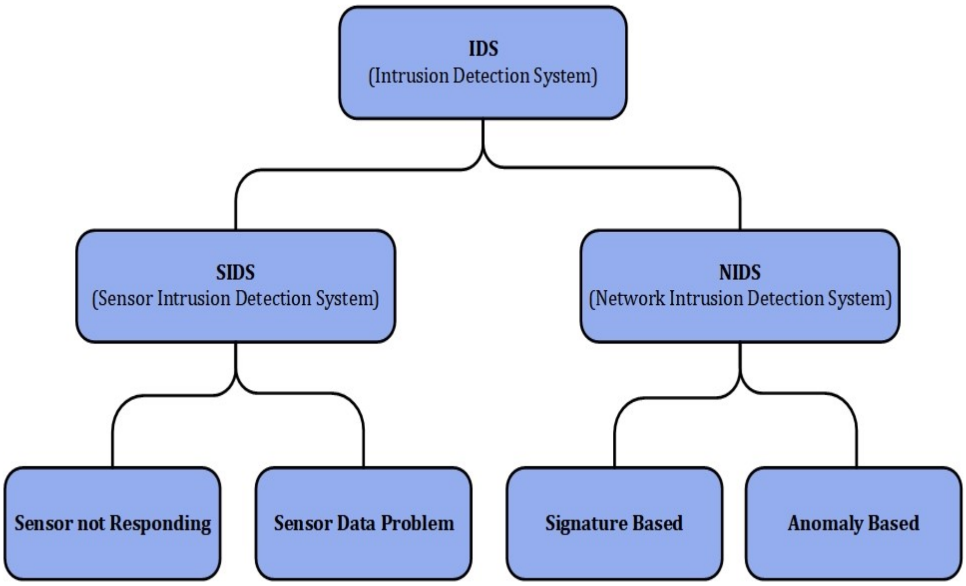

IDS
(Intrusion Detection System)
SIDS
(Sensor Intrusion Detection System)
NIDS
(Network Intrusion Detection System)
Sensor not Responding
Sensor Data Problem
Signature Based
Anomaly Based

Figure 3.1: Proposed IDS

3.2.1 IDS

IDS is the cyber security defensive tool which plug over the network in monitor mode so it can inspect all incoming and outgoing network traffic and, on the basis of network traffic it can predict the anomalies. We proposed the IDS for Edge computing so it can usable on Edge servers for the protection of end IoT devices from different possible attack. In the start IDS only have the signature base technique to find the intrusion so this type of IDS gets fail when attack come with the new signature because new signature is not being updated in the IDS signature library. So next generation of IDS are being introduced which able to check the network traffic trends and incase of miss match IDS generate the alert for the intrusion but these are not enough to ensure proper security.

IDS also have different sub type like NIDS HIDS etc. Each type of IDS having some respective use and limitations. So we combine two types of IDS which also have different sub components. Currently there is a time of next generation IDS those not only detect the intrusion in anomaly or normal traffic but also give details about the which class type of attack are being captured. IDS we suggest is also able to communicate with the other security solution so it can update the signature in other solution like IPS, firewall or ACL to block traffic from specific address.

3.2.2 SIDS

This is the subtype of IDS we used in building our IDS. SIDS is basically a sensor intrusion detection system which is designed to detect the sensor's intrusion. It has two main components which can be built using different techniques. We also made the prototype by coding the sensor intrusion detection system, which basically monitors the state of connection and inspects the sensors data to ensure sensors are sending the right data. To understand both of the features, we discuss them with an example. Sensors that are disconnected from the network are detected as intrusions by the sensor not responding module, which generates an alert for the system administrator to deal with this anomaly.

Without this module checking the state of senor whether it is online or offline is not possible. There are different possible ways to implement this module, but for the sake of concept we implement this module by using the ping and technique which monitors the connection state of the sensor. Sensor Data Problem is actually the second module of SIDS which inspects the sensor's data. Like we have a temperature sensor and we have configure some rules for that sensor in SIDS.

That the first rule checks the range value of temperature in between maximum and minimum limit. so if sensor does not generate the data between those limits SIDS detect this as an anomaly. The second rule is for check the format of data so If sensor sends the data in the wrong format, like temperature data must be in decimal numbers and sensor starts sending in floating point than again SIDS detects this as anomaly and alerts the system administrator.

If sensors directly communicate on https with the cloud than we have to provide the certificate to IDS for decryption of traffic and detection of such attacks. Both of these modules are important because these provide clear visibility of the network otherwise we are unable to see what's going on the network which sensor is off or which sends invalid data.

3.2.3 NIDS

The NIDS is Network Intrusion Detection System which we designed to protect the network from different possible intrusions by detecting them timely and adding the mitigation rule. Our proposed NIDS is designed in a hybrid mode, where one works as a signature-based system, while the other acts as an anomaly-based system. Signature base just match the IoC signatures which are present in the database having different forms like Hash value, IP address, URL, etc. If signature gets match IDS alerts for the intrusion and in case of Interconnected Threat Mitigation adds rule in the other security solution as well.

If we don't have the signature, it goes to an anomaly detection engine which predicts on the basis of a trained model if an attack is detected, then it updates the signatures in the database and other security solutions. That's how we don't consume too many resources and propose a light weight anomaly detection engine that can be built on edge servers. Because we have some limitations in the anomaly based engine so we full fill gap using the signature base engine as well.

3.2.4 Interconnected Threat Mitigation

The term is extracted from Connected Threat Dssefense which is the concept of enterprise security solution companies. In Interconnected Threat Mitigation our IDS has the feature to share the IoCs with some other security tools such as Firewall for the mitigation of attack. The very clear example of this feature is DoS attack. If an IDS detects a DoS attack, then IDS updates the blocklist IP in the Firewall so it can block the traffic from the attacker side so we can block the attack automatically. Same as for Signature and IoCs update we need Threat Intelligence solutions which can be integrated with IDS to automatically update IDS for latest threats signatures and IoCs. So due to such components we said our proposed IDS will have interconnected Threat Mitigation.

3.2.5 Proposed Solution Architecture

Here we discuss the proposed system architecture in which we are improving and try to suggest an IDS which is smart enough to detect modern intrusions on edge computing and as light weight as possible. The diagram is presented in form of flowchart. First of all we have network traffic which is passed to IDS, then IDS separates the traffic for sub modules. The traffic carrying the sensors data sends it to SIDS module which passes it to SIDS rules engine. If data breaks the policy, then the system alerts this as sensor intrusion otherwise as normal traffic. Network traffic is passed to signature base engine which matches traffic patterns with the present signatures and in case of match hit an alert otherwise passes to anomaly base engine which has a model trained on modern datasets to predict the modern attack.

If engine detects traffic as anomaly then it simply hits an alert and if some other security solution is integrated, the proposed system pushes the mitigation policy in it, otherwise leave it as normal traffic. This is how the proposed system will work on the proposed architecture. We try to made engines as much lighter to run on small resources but we are presented a concept. Actually, testing of the system on different chips is also an complete research which will be continue as feature work. Optimization of engines and use of latest techniques is also research area for the components.

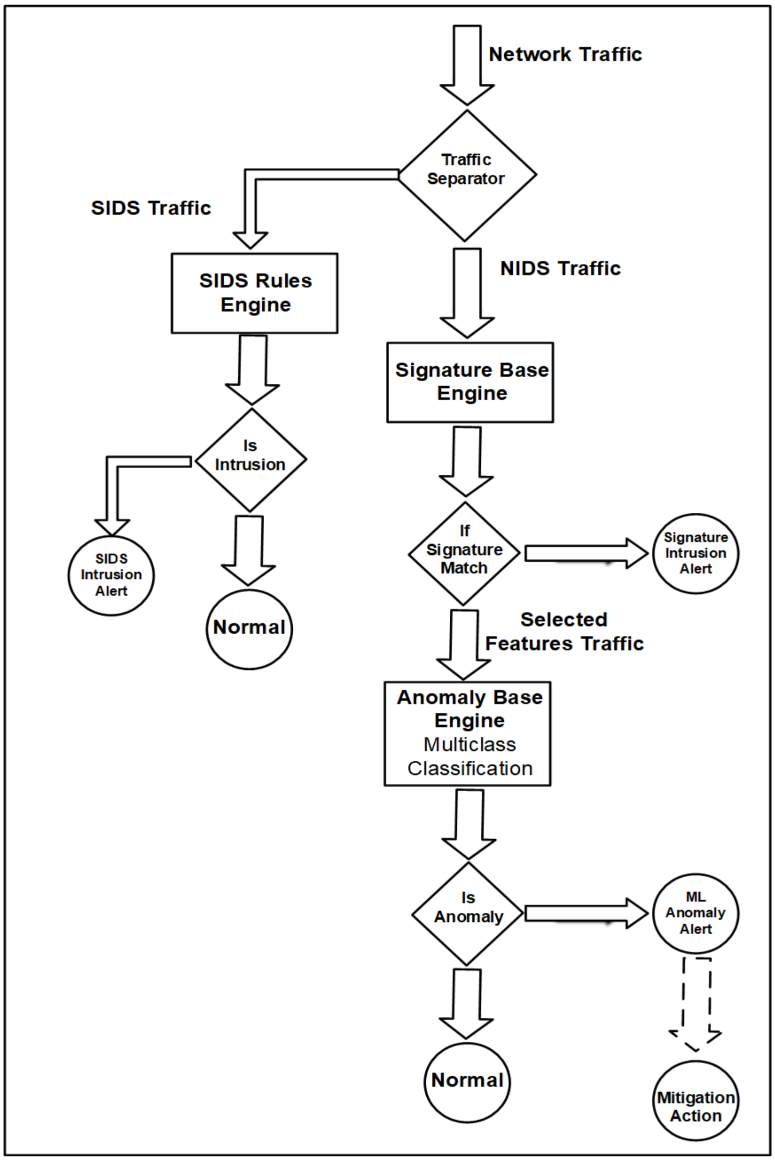


Figure 3.2: Proposed IDS Solution Architecture

3.2.5.1 Signature Based Engine

There are many techniques to implement the signature base engine. But here because this implementation of signature base engine in not a part of our research we are just proposed the use of already developed and best possible engine. The internal architecture of a signature based engine is shown below.

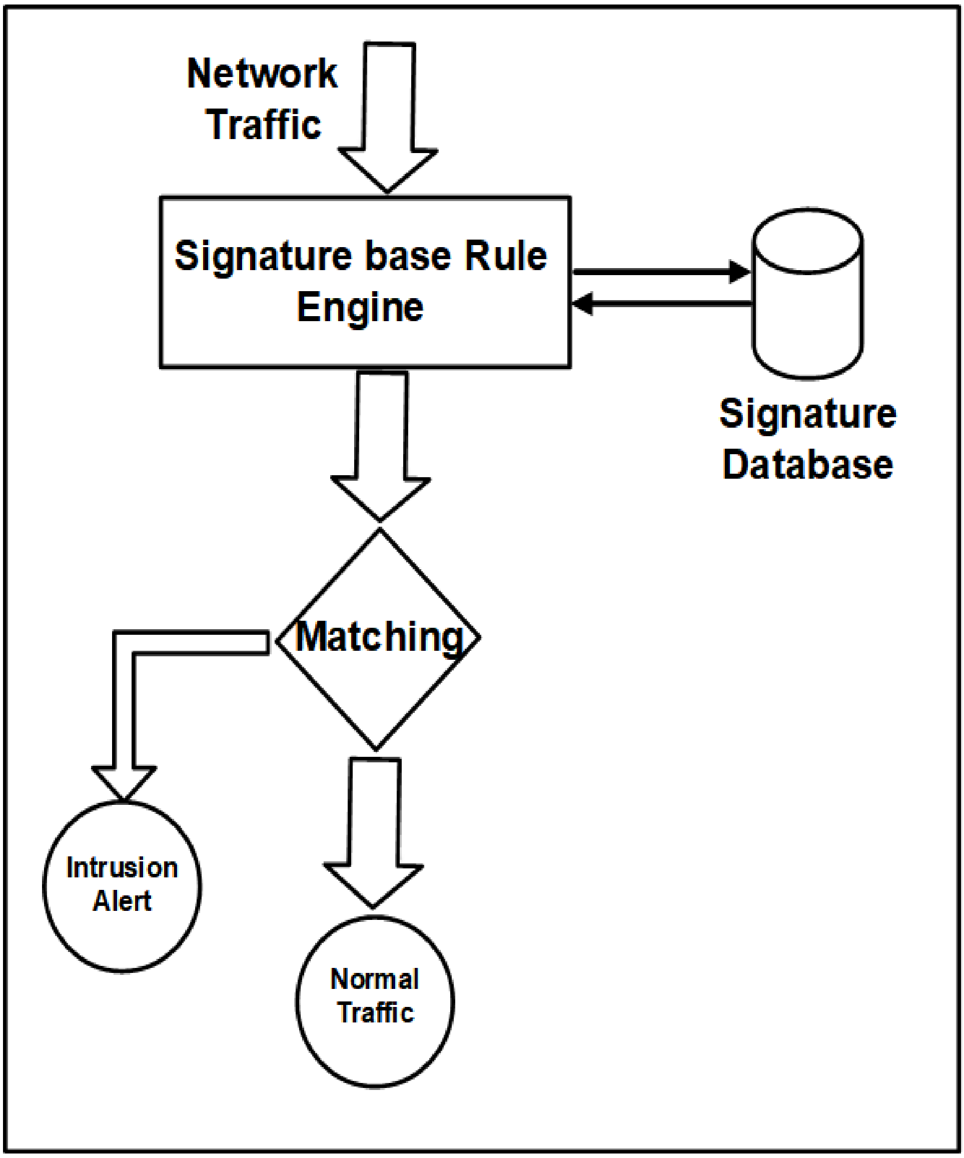


Figure 3.3: Signature Based Engine

3.2.5.2 Anomaly Based Engine

Anomaly base engines use machine learning and artificial intelligence to decide about the intrusion. The basic architecture of anomaly base engine is shown below Fig 3.4. We are also researching on building an anomaly base engine which use the latest dataset and machine learning to decide about the intrusion. We use it as the module which work with other different modules to detect the intrusion. First of all system apply signature based comonent because ML is resource intensive so ML can be apply in only case when no signature is being matched.

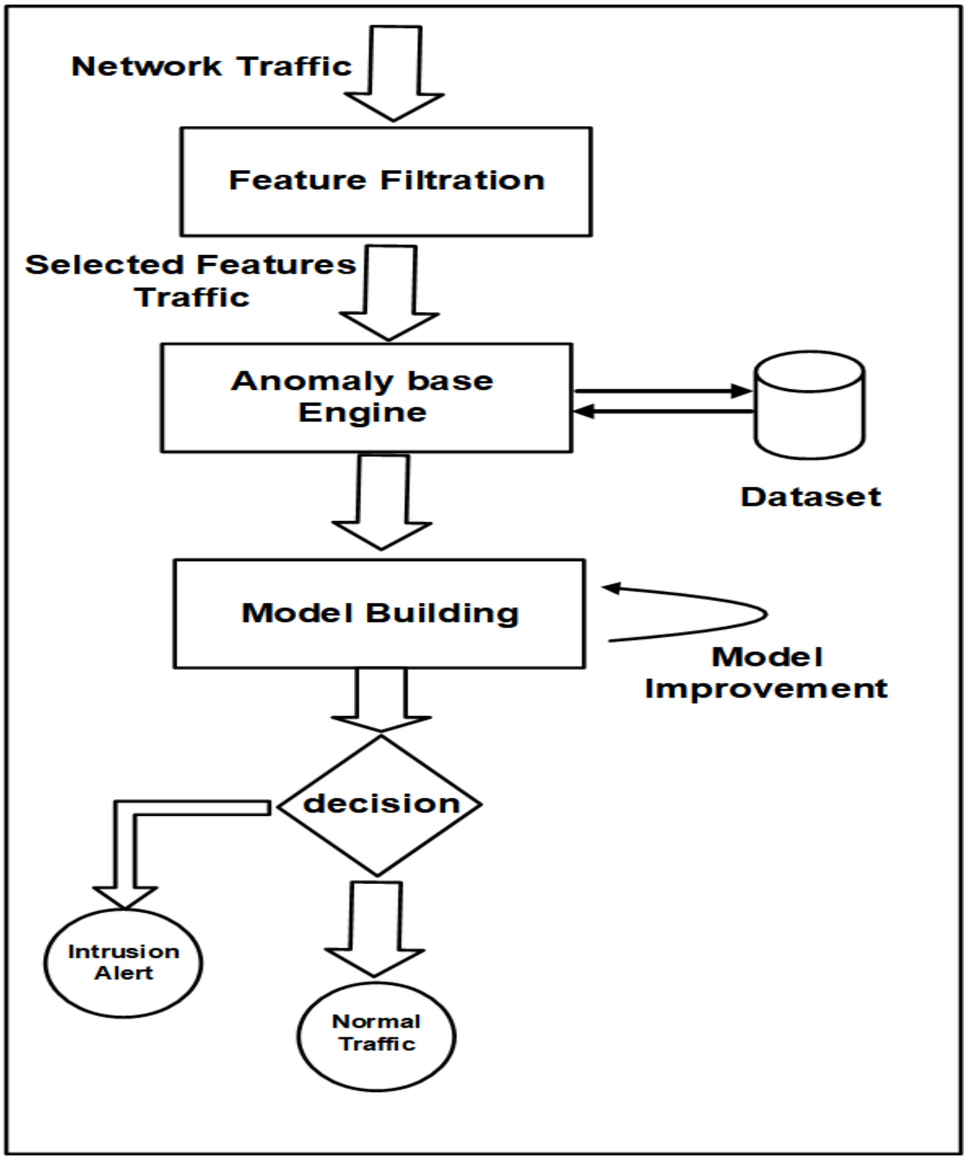


FIGURE 3.4: Anomaly Based IDS Architecture

3.2.6 Proposed Solution Deployment Architecture

The deployment architecture of the proposed system is very simple. The basic need is a network. There is no concept of Edge computing in the old architectures, and sensors send their data directly to the cloud. This architecture is a security issues because of limited resources in sensors. They send the data in a plan so Man in the Middle attacks is possible. Due to direct connection, the attacker directly attacks the sensors to launch the DoS or DDoS attack which causes service interruption etc. Classical Structure is also shown in below Fig 3.5.

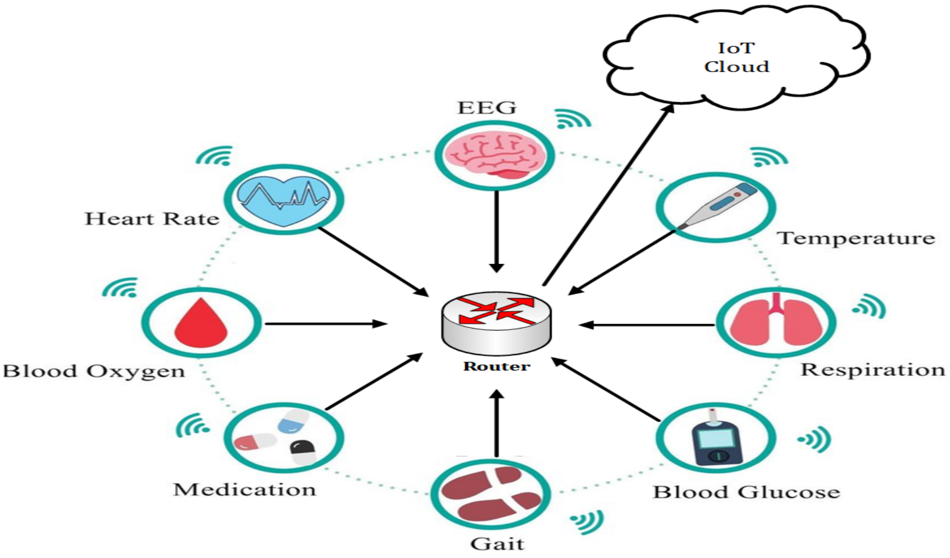


FIGURE 3.5: Previous Architectures without Edge Server

Incase of Edge server available still there is no security solution so edge server might apply the encryption before sending the data to mitigate the Man in the Middle attack but other attacks are still possible. System architecture with the Edge computing concept is also shown below Fig 3.6

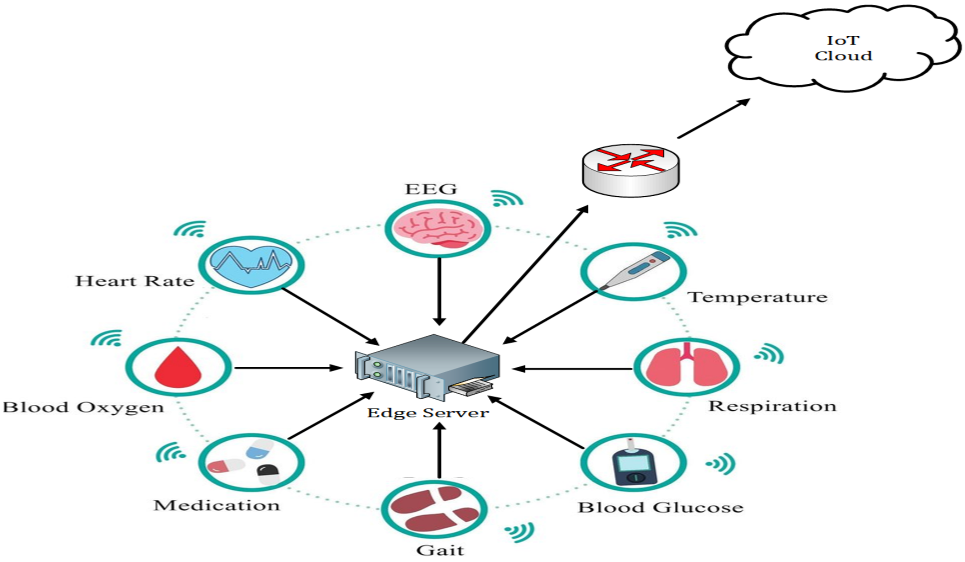


FIGURE 3.6: Previous Architectures with Edge Server

Our proposed IDS is installable on the edge Server which can be PC or some workstation. We design solution as optimized but still there is need to test it on the small devices like Raspberry Pi but in future, we optimize this to chip level deployment. The deployment diagram of proposed shown below Fig 3.7.

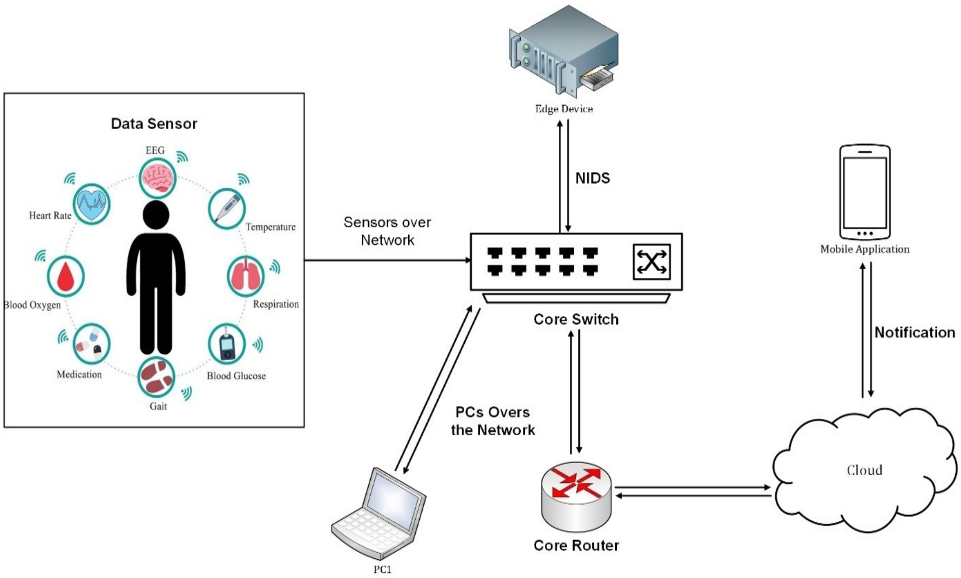


Figure 3.7: Proposed Deployment Architectures with Edge Server

IoT devices Central control mostly also link with the cloud for enable access from anywhere in the world so after processing or decision at edge server it also update the same state on cloud. So in deployment architecture there is internet and cloud from which mobile applications design to view the IoT devices states for system is implemented. Sensors are connected to edge server via core switch and sends their data on it for initial data processing and syncing the states to cloud. Core switch is also directly connected to the core router which provide the facility of external routing on internet for cloud communication on internet. Other internal system is also possible to share the same network so they might directly connect to core switch or connect via some different access switches.

Due to this architecture every traffic is to being pass through the core switch and we mirror the network traffic to edge server for IDS which detect the possible intrusions. Sensors are to be integrated with the IDS for SIDS module on edge server so SIDS also inspect sensors data and validate it on define rule set. This is the basic architecture we can also place other security solution here on their respective places for an example if we also want to place the Firewall between the core router and core switch than Firewall is able to block the traffic. IDS is connected to Firewall than if IDS detect the intrusion from some external of internal IP than IDS push the block rule policy for respective traffic in Firewall than Firewall block the traffic. That's how deployment architecture helps system to protect the network and mitigate the attack.

3.2.7 Proposed Solution Working Scenario

There are multiple working scenarios but we discuss the important one.

Case 1: This is the case when a hacker comes from the external network and launches some cyber attack on the system. He comes from the internet and then reaches the core router. The main purpose of the router is to route the traffic between the networks so it routes the traffic without any security check so legitimate and illegitimate traffic pass through it. In some new routers, there is a concept of ACL that could be defined by the administrator to drop traffic from specific IP but this get fails when attackers come from the new IP address. When an attack pass through the core router than it reaches to core switch. when the attacker performs suspicious activity if the IDS has the signature for the activity then IDS detects it via the signature base engine otherwise IDS passes it to ML where IDS predicts it as malicious and then alert the administrator. The attacker gets success in his attack when ML detects the attack as legitimate traffic therefore, multiple different security solutions are needed for enabling multi-dimensional and multi-layer security.

Case 2: When there is an insider threat. In the same way as the first case, if an insider attacks the network sensor or attempts to perform some sort of attack, the IDS will also detect him and issue an alert for the administrator.

Case 3: In case a sensor is damaged or sends an invalid value, the IDS also inspects the sensor's data on the basis of defined rules. If there are any problems then the IDS generates an intrusion alert. Sometimes attackers launch sensor spoofing attacks in which their deployed sensors act as actual sensors and in times of attack these sensors might produce the wrong formatted or invalid data, which impacts the effectiveness of the system.

3.3 Research Methodology

We discovered some research gaps after completing the literature review, which led to some research questions. In order to answer those questions, we use a combination of experimental and survey research methodologies. As a result of the surveys, we identify the different previous systems and the problems associated with them. Our next step is to combine different research solutions to answer our research questions, then to improve and evaluate the results, we conduct experiments with different combinations and settings, and conclude the results discussed in the next chapter. In our research topic we are proposing an IDS for edge computing that further proposed a suitable deployment architecture and improvement in the previous machine learning model. So, for improvement in architecture, we study the different surveys and for machine learning engine we firstly select the dataset among the different datasets than at last randomly selected algorithms applied on dataset. At last, we proposed the best algorithm which gives high accuracy in low time.

3.4 Data Collection

For conduct research and building the anomaly-based engine we studied the multiple datasets than select among the one. Different datasets build in different context and technology. Some are extracted from the real environments and some from the specific attack simulations. We select the dataset which is modern and useable for detection of modern attacks. The flow chart of methodology is shown below as well.

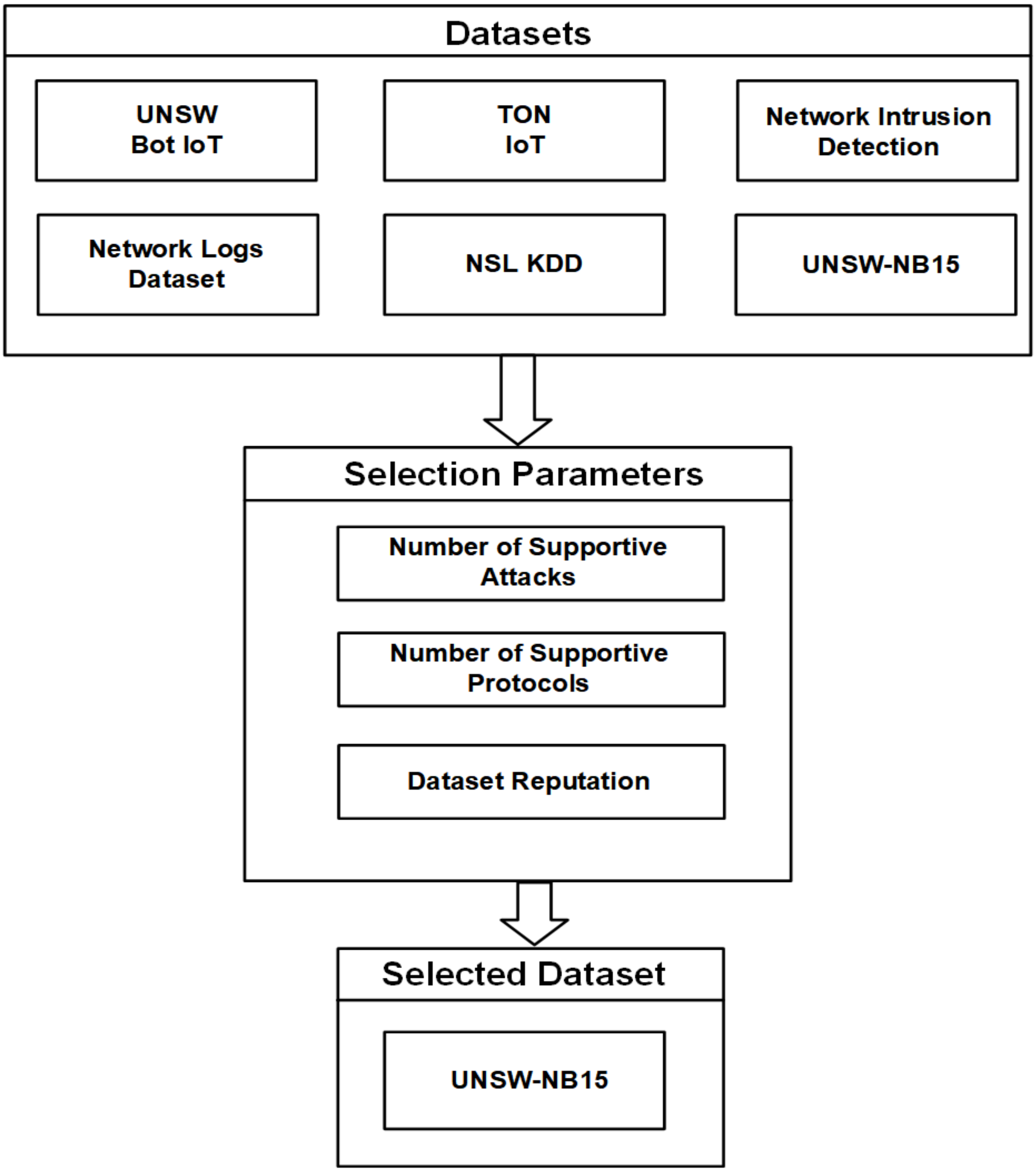


FIGURE 3.8: Data Collection

We filter the modern dataset on 3 different parameters such as number of attack classes in the dataset, how much protocols traffic in the dataset and reputation of the dataset which taken from literature survey. After the selection of dataset, we select the machine learning algorithms randomly which are commonly use for building the anomaly based IDS. Then we perform the different operations on dataset than evaluate the accuracy. Before further discussion on methodology, I will discuss the previous dataset comparison with selected dataset and answer the question for why the dataset is being selected.

Table 3.1: Datasets with respect to supported Protocols and Attack Classes

No	Dataset	Protocols	Attack Classes
1	UNSW-NB15	• 130 Protocols 3PC, a/n, aes-sp2-d, Any, argus, aris, arp, ax.25, bbn-rcc, bna, br-sat-mon, cbt, cftp, chaos, Compaq-peer, cphb, cpnx, crtp, crudp, dcn, ddp, ddx, dgp, egp, eigrp, emcon, encap, eithercap, fc, fire, ggp, gmtp, gre, hmp, iatp, ib, idpr, idper-cmtp, idrp, ifmp, igmp, igp, il, inlsp, ip, ip, ipcomp, ipcv, ipip, iplt, ipnip, mobile, mtp, mux, narp, netblt, nsfnet-igp, nvp, ospf, pgm, pim, pipe, pni, pri-enc, prm, ptp, pvp, qnx, rdp, rsvp, rvd, rcp, tlsp, tptt, trunk-1, trunk-2, ttp, udp, unas, uti, vines, visa, rmtp, vrrp, wb-expak, we-mon, wsn, xnet, xns-idp, xtp, zero	• 10 Classes Analysis, Backdoor, DOS, Exploits, Fuzzers, Generic, Normal, Reconnaissance, shellcode, worm

2	UNSW BoT IoT	• 5 Protocols Arp, icmp, ipv6-icmp, tcp, udp	• 4 Classes DDOS/DOS, Normal, Reconnaissance, Theft
3	TON IoT	• 3 Protocols Icmp, udp, tcp	• 9 Classes Backdoor, mirtm, dos/ddos, normal, injection, password, ransomware, scanning, xxs
4	Network Intrusion Detection	• 3 Protocols Icmp, udp, tcp	• 2 Classes Anomaly, Normal
5	NSL KDD	• 2 Protocols tcp, ping	• 5 Classes DOS, Normal, Prob, R2L, U2R
6	Network Logs Dataset	• 5 Protocols Arp, icmp, ipv6-icmp, tcp, udp	• 5 Classes Normal, UDP-Flood, Smurf, SIDDOS, HTTP-Flood

After the detail analysis we have found that UNSW-NB15 is the dataset which supports the maximum number of protocols with maximum attack classes. If we talk about the reputation of dataset which is already good and being used in building IDS previously. We can also use this dataset in our research work for building the general NIDS module which detect the common possible network attacks. UNSW-NB15 is created by the PerfectStrom tool named as IXIA in Cyber Range Lab of Australian Center of Cyber Security(ACCS) in sence of normal and synthetic contemporary attack behaviour. Tcpdump tool utilized for capture raw traffic. The number of records in the training set is 175,341 records and the testing set is 82,332 records from different the types of attack and normal. Before going to next topic we discuss the attribute features are available in UNSB-NB 15. Features are discussed below

TABLE 3.2: UNSW-NB 15 Dataset Features explain

No	Name	Type	Description
1	srcip	nominal	Source IP address
2	sport	integer	Source port number
3	dstip	nominal	Destination IP address
4	dsport	integer	Destination port number
5	proto	nominal	Transaction protocol
6	state	nominal	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	dur	Float	Record total duration
8	sbytes	Integer	Source to destination transaction bytes
9	dbytes	Integer	Destination to source transaction bytes
10	sttl	Integer	Source to destination time to live value
11	dttl	Integer	Destination to source time to live value
12	sloss	Integer	Source packets retransmitted or dropped
13	dloss	Integer	Destination packets retransmitted or dropped
14	service	nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service
15	Sload	Float	Source bits per second
16	Dload	Float	Destination bits per second

17	Spkts	integer	Source to destination packet count
18	Dpkts	integer	Destination to source packet count
19	swin	integer	Source TCP window advertisement value
20	dwin	integer	Destination TCP window advertisement value
21	stcpb	integer	Source TCP base sequence number
22	dtcpb	integer	Destination TCP base sequence number
23	smeansz	integer	Mean of the how packet size transmitted by the src
24	dmeansz	integer	Mean of the how packet size transmitted by the dst
25	trans_depth	integer	Represents the pipelined depth into the connection of http request/response transaction
26	res_bdy_len	integer	Actual uncompressed content size of the data transferred from the server's http service.
27	Sjit	Float	Source jitter (mSec)
28	Djit	Float	Destination jitter (mSec)
29	Stime	Timestamp	record start time
30	Ltime	Timestamp	record last time
31	Sintpkt	Float	Source interpacket arrival time (mSec)
32	Dintpkt	Float	Destination interpacket arrival time (mSec)
33	tcprtt	Float	TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'.

34	synack	Float	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
35	ackdat	Float	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
36	is_ftp_login	Binary	If the ftp session is accessed by user and password then 1 else 0.
37	ct_ftp_cmd	integer	No of flows that has a command in ftp session.
38	is_sm_ips_ports	Binary	If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0
39	ct_state_ttl	Integer	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
40	ct_flw_http_mthd	Integer	No. of flows that has methods such as Get and Post in http service.
41	ct_srv_src	integer	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
42	ct_srv_dst	integer	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43	ct_dst_ltm	integer	No. of connections of the same destination address (3) in 100 connections according to the last time (26).

44	ct_src_ ltm	integer	No. of connections of the same source address (1) in 100 connections according to the last time (26).
45	ct_src_dport_ltm	integer	No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).
46	ct_dst_sport_ltm	integer	No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
47	ct_dst_src_ltm	integer	No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).
48	attack_cat	nominal	The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
49	Label	binary	0 for normal and 1 for attack records

3.5 Machine Learning Algorithms Selection

Selection of the machine learning algorithm is an important thing because this plays the vital role in achieving accuracy. There are much machine leaning but we select randomly usually use for building IDS. Major selected algorithms are from supervised machine learning and other from unsupervised. Selected algorithms are discussed below

3.5.1 Random Forest

The training phase of random forests, an ensemble learning approach for classification, involves the construction of several decision trees, each of which is then used to predict an output class. Due of their tendency to overfit to their training set, decision trees are improved by replacing them with random choice forests. The time complexity and formula is being shown below

Time Complexity= O(T ·D)

$$RFfi_i = \frac{\sum_{j\in \text{ all trees }} \text{normf } i_{ij}}{T}$$

- RFfi sub(i) = the importance of feature i calculated from all trees in Random Forest model
- normfi sub(ij) = the normalized feature importance for i in tree j
- T = total number of trees

3.5.2 Decision Tree

A decision tree is a structure similar to a flowchart in which each internal node represents a ”test” on an attribute (for example, whether heads or tails is the result of a coin flip), each branch reflects the results of the test, and each leaf node represents a class label (decision taken after computing all attributes).

Time Complexity= O(m · n2)

3.5.3 SVM (Support Vector Machine)

Support vector machines, often known as SVMs, are a type of algorithm used in machine learning that performs analysis of data for classification and regression. The Support Vector Machine (SVM) is a type of supervised learning that analyses data and places it in one of two categories. An SVM generates a map of the data after it has been sorted, with the margins between the two being as far apart as is practically possible. Heavy reliance on it for use with Intrusion Detection Systems.

Time Complexity= between O(n$\hat{2}$) and O(n$\hat{3}$) with n the amount of training instances

3.5.4 NaïveBays

In the field of machine learning, naive Bayes classifiers are a family of straightforward ”probabilistic classifiers.” These classifiers are created by using Bayes' theorem while making strong (naive) assumptions of independence between the features. Time complexity and basic formula is shown below

Time Complexity=O(d*c) where d is the query vector's dimension, and c is the total classes

$$\mathrm{P}(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

- P(A) is the Probability of A occuring.
- P(B) is the Probability of B occuring.
- P(B | A) is the Probability of B occuring given evidence A has already occured.
- P(A | B) is the Probability of A occuring given evidence B has already occured

3.5.5 KNN

The k-nearest neighbours (KNN) method is a basic, easy-to-implement supervised machine learning technique that may be used to tackle both classification and regression problems. Its name comes from the fact that it finds the neighbors that are the closest to each data point. The k-nearest neighbor classifier fundamentally relies on a distance metric. The better that metric reflects label similarity, the better the classified will be. The most common choice is the Minkowski distance The time complexity and Formula is being shown below

Time Complexity= O(n)

$$\operatorname{dist}(\mathbf{x}, \mathbf{z}) = \left(\sum_{r=1}^{d} |x_r - z_r|^p \right)^{1/p} .$$

3.5.6 Logistic Regression

Logistic regression is a statistical model that, in its most fundamental form, employs the application of a logistic function to model a binary dependent variable. There are, however, numerous extensions of this model that are far more complicated. Estimating the parameters of a logistic model is the purpose of the regression technique known as logistic regression (also known as logit regression). The time complexity and formula is being shown below

Time Complexity= O(n.d)

$$g(E(y)) = \alpha + \beta \times 1 + y \times 2$$

Here, $g()$ is the link function, $E(y)$ is the expectation of target variable and a $+\beta \times 1 + \gamma \times 2$ is the linear predictor (α, β, Y to be predicted). The role of link function is to 'link' the expectation of y to linear predictor.

3.5.7 MLP(Multi-Layer-Perceptron)

It is a Feed Forward Neural Networks. A feedforward neural network (FNN) is an artificial neural network wherein connections between the nodes do not form a cycle. It is Used in Research for Intrusion Detection. Formula of MLP is being shown below

$$y = f(WxT + B))$$

- f is the activation function
- W is the set of Parameter
- x is the input vector
- b is the basic vector

3.6 Dataset Preprocessing

In order to improve accuracy after selecting a dataset for machine learning, we need to perform some pre-requisite operations over it. The first step is to clean the dataset by removing all the irrelevant rows or rows with garbage data, and then normalize the data. Dataset normalization is reffers to the organization of data to appear similar across all records and fields. The goal of normalization is to transform features to be on a similar scale. This improves the performance and training stability of the model. In the end, we select the best feature that plays a role in achieving the highest accuracy possible. So we see that all the features play a role in achieving high possible accuracy. It seems that only the id column, which is used to uniquely identify the column, is useless, so we drop it, and it doesn't affect the accuracy, but all other features play a role, so dropping an individual feature leads to a decrease in accuracy. so we use all those features in our machine learning engine. The flow is also presented in diagram below

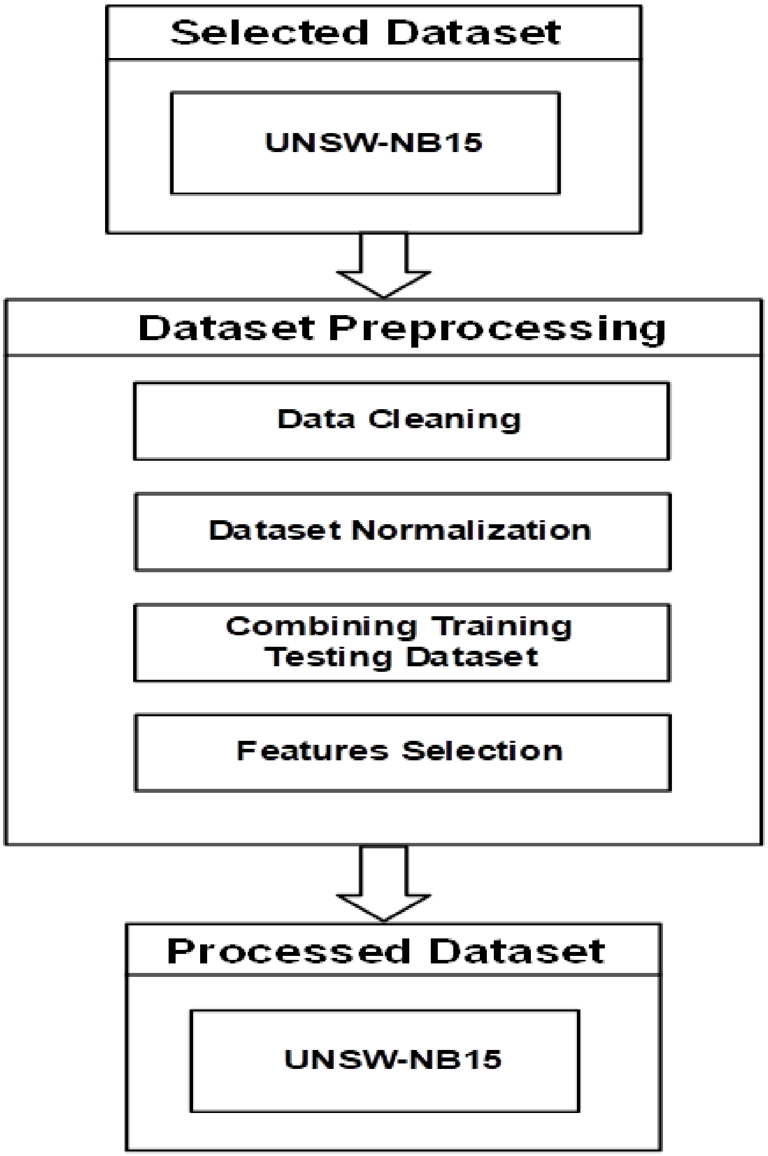


Figure 3.9: Data Preprocessing

3.7 Methodology of ML based NIDS Engine

When processed data set is ready, we pass the processed dataset to selected machine learning algorithms which generate the results accordingly. When all algorithm gives their results so we suggest the algorithm which take a small time and gives the highest accuracy. Then in results section we compare the result of each algorithm in detail and select the final one.

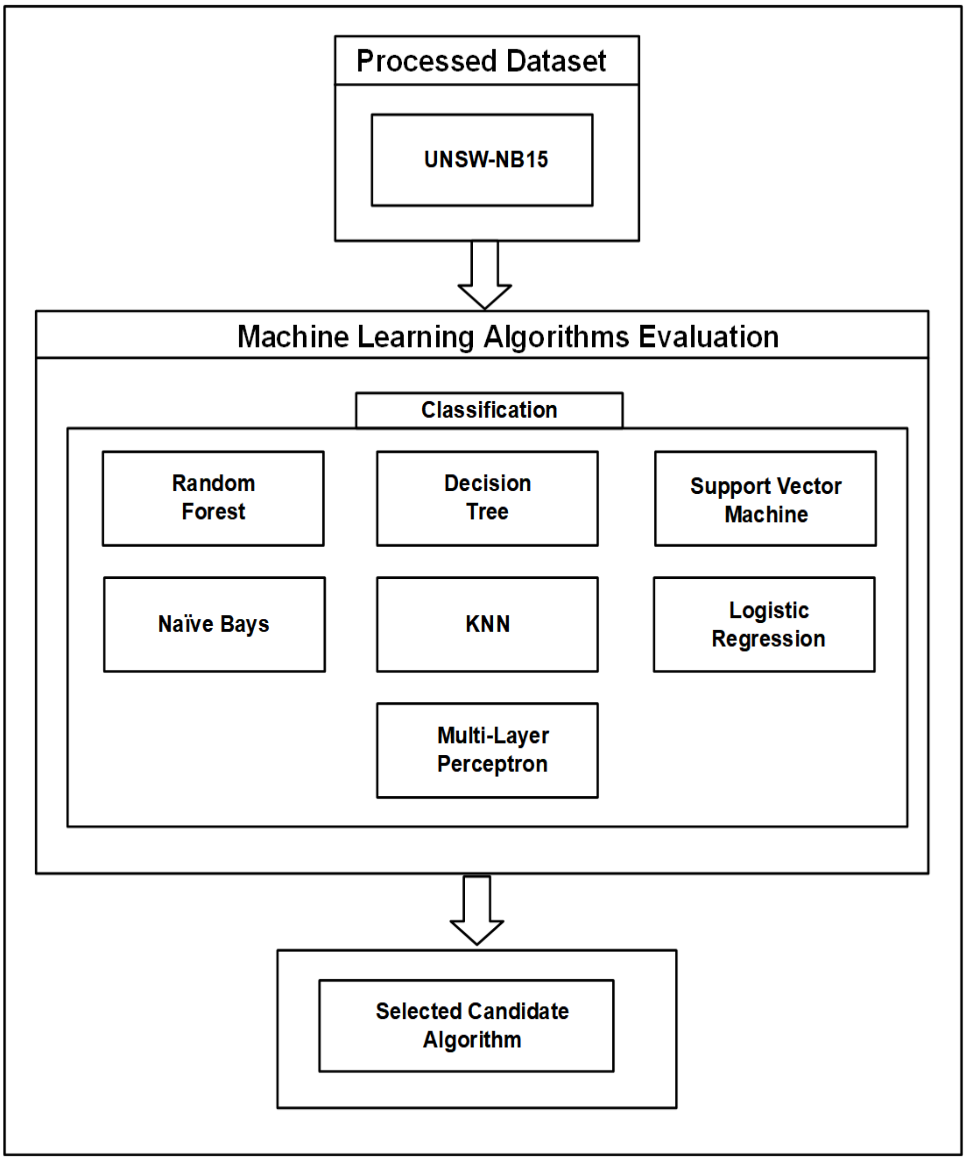


FIGURE 3.10: Selection of Candidate Algorithm

3.8 Methodology of rule based SIDS Engine

Proposed SIDS engine is based on the specific rules which we define for the particular data. This module check the logs and monitor data send by the sensors. That's how this module provide the state visibility of the network like which machine or sensors is down or up and which causing the source data intrusion. First of all data is passed to SIDS engine in which we have two different sub engines are working. The Data is passed to rules engine which inspect the data on the bases of define rules and if there is problem in the data system generates an alert.

Active engine is monitoring the logs and connection status of system and sensors if the sensor close the connection or the activity logs get older than system pings the sensor after that on the bases of results system generates an alert. Some sensors make connection and work on TCP where as some work without making connection those sensors use UDP. System simply check the log for these type of sensors.

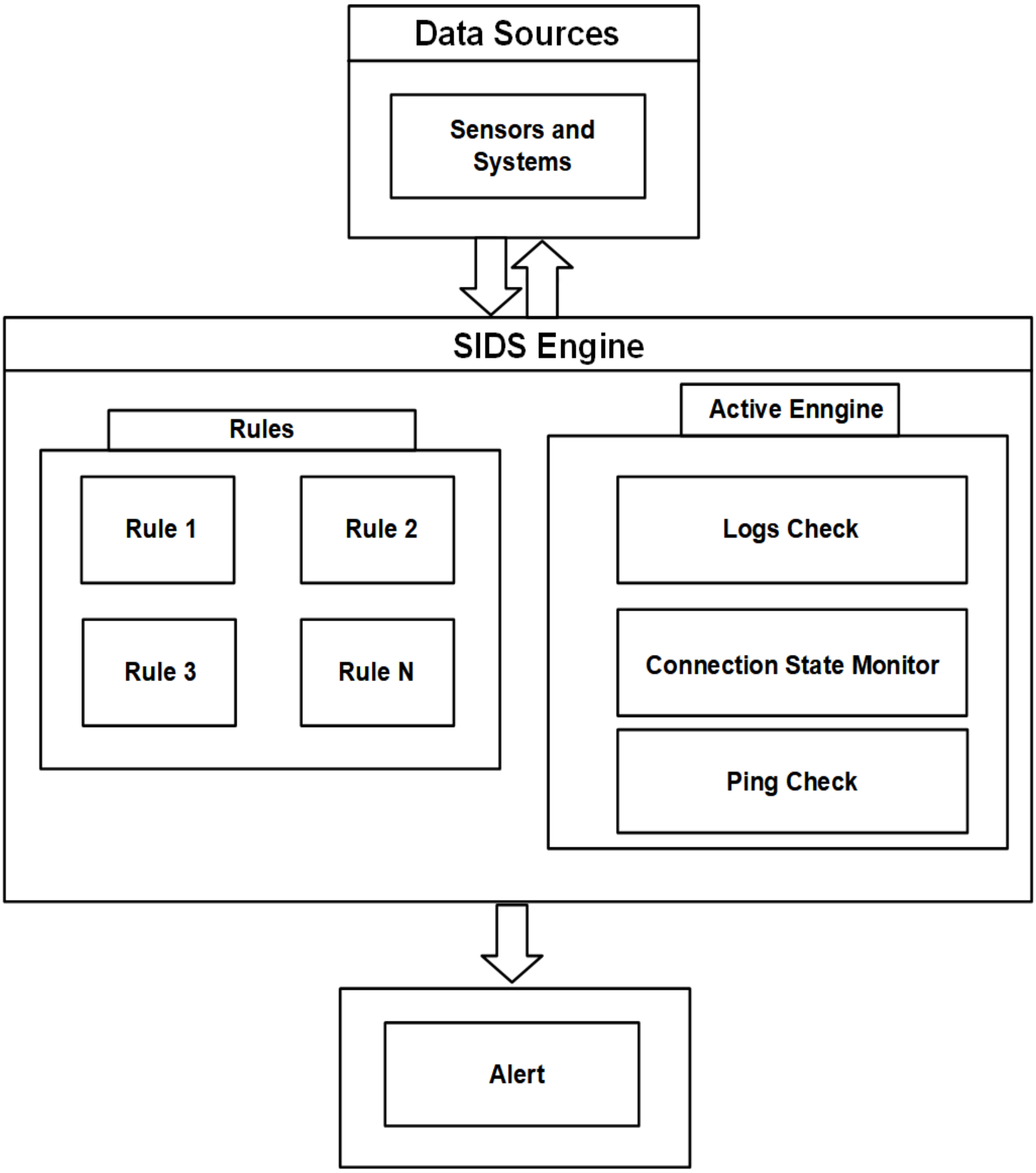


FIGURE 3.11: Methodology of rule based SIDS Engine

3.9 Evaluation

After defining the research methodologies, we conduct the experiments, evaluation of experiments are important to discuss because of bases of evaluation we suggest the right algorithm to use. For the evaluation of experiments, we find the accuracy of each algorithm than macro avg and weighted avg for precision, recall and f1score. These all re base on the values of TN: True Negative FP: False Positive FN: False Negative TP: True Positive

3.9.1 Accuracy

The number of accurate predictions that your model was able to make for the entire test dataset is referred to as its accuracy. The following equation is used to determine its value

$$\text{accuracy } = \frac{TP + TN}{TP + TN + FP + FN}$$

3.9.2 Precision

The degree to which a positive prediction is accurate is referred to as its precision. To put it another way, it means that if a result is predicted to be positive, how certain are you that the result will in fact be positive? It is determined by applying the formula that is as follows:

$$\text{precision } = \frac{TP}{TP + FP}$$

3.9.3 Recall

The recall rate, also known as the true positive rate, is a metric that determines how many true positives are predicted out of the total number of positives in a dataset. In certain instances, it is also referred to as the sensitivity. The following equation is used to calculate the value of the measure:

$$\text{recall} \ = \frac{TP}{TP + FN}$$

3.9.4 F1-Score

The F1-score is the F-score that is utilized the vast majority of the time. It is a mixture of precision and recall, such as the harmonic mean of both of them. The formula below can be used to get an individual's F1 score:

$$F1 = 2 \cdot \frac{\text{precision.recall}}{\text{precision} \ + \ \text{recall}}$$

3.10 Tools and Technologies

To perform the complete experiments and suggestions of models we use the different tools and technologies mention below

- Microsoft Visio – use for creation of diagrams
- Microsoft Excel – use for manage the dataset
- Python - For Programming of SIDS and Machine Learning
- Sklearn – Machine Learning algorithms library
- PyCharm – IDE for Python Programming
- Latex – For Thesis Writeup
- Weka – For resampling and Initial experiments

3.11 Conclusion

In literature review, research work conclude that cyber war is the 5th generation of warfare. Cyberattacks are becoming increasingly sophisticated, so organizations needs proper security solutions that provide strong defense and detection capability. The very first phase is to detect the attack, then to decision about how to mitigate it. Different attacks use different strategy. Research work pimproves research work from [2] and implementing IDS in the edge server. Furthermore, research work proposed a SIDS engine and a complete deployment architecture, which resulted in a good accuracy rate. We still need to test the proposed solution on small edge servers, which will be the focus of our future research. In addition, research work provide complete information about the tools and technologies used in the evaluation.

Chapter 4

Experiments, Results and Evaluations

This chapter describes the experiments performed for our research work, followed by a discussion and demonstration of the results. As part of the experiment, research work also evaluated the results. An experimental setup is developed on the basis of a proposed system and selected methodology. Our next step is to gather the results of each experiment, which will be discussed further in order to determine the best one. For the motivation and initial start, we use the Weka tool for perform the different experiments in which we apply different machine learning algorithms and then we see the results for each. Due to some limitation in the tool, we move towards the self-programming in which we write the program for each algorithm and get the results. Details discussed below.

4.1 Experimental Setup

This section explain the complete setup which we build for conduct different experiments. We take and a computer machine which have the following resources

- Intel Core i-7 5500U 2.50 GHz 2 core Processor

- 16 GB of DDR3 Ram

- 256 GB SSD

Then research work collect the different datasets and install the Weka tool which is the automated machine learning tool we just need to select the dataset and the algorithm than tool provide us the results. We apply this tool over the all dataset and gather different results. This is for the testing and start motivation because Weka is automated tool which cannot be modified.

Then due some limitations of algorithms and slow speed of Weka we move towards the use of Google Colabs but machine learning algorithms take much time which is not available in free version of google colabs so we move towards the coding.

For sake of coding we install the python engine which provide the facility to run python codes, for writing code we install and use PyCharm 2022.1. For machine learning algorithms we use the Sklearn library. That's how we simply make a experimental setup we write all the algorithms in same file and then apply each algorithm to same dataset one by one and copy the results to other document.

4.2 Initial Experiment using Weka

It is a tool which have the different machine learning algorithms use for data preparation, classification, regression and clustering. We run the Weka on different datasets but here we discuss the Weka on UNSW-NB15. We apply the six different algorithms Decision Table, j48, Naïve Bays, Random Forest, SMO, ZeroR. Due to problem in tool we are not able to assign the separate training and testing data. So we just apply the algorithm on training dataset and evaluate using 10- Fold cross validation. These experiment perform on training dataset without any change in it. In Fig 4.1 below you can see Random Forest give the highest accuracy.

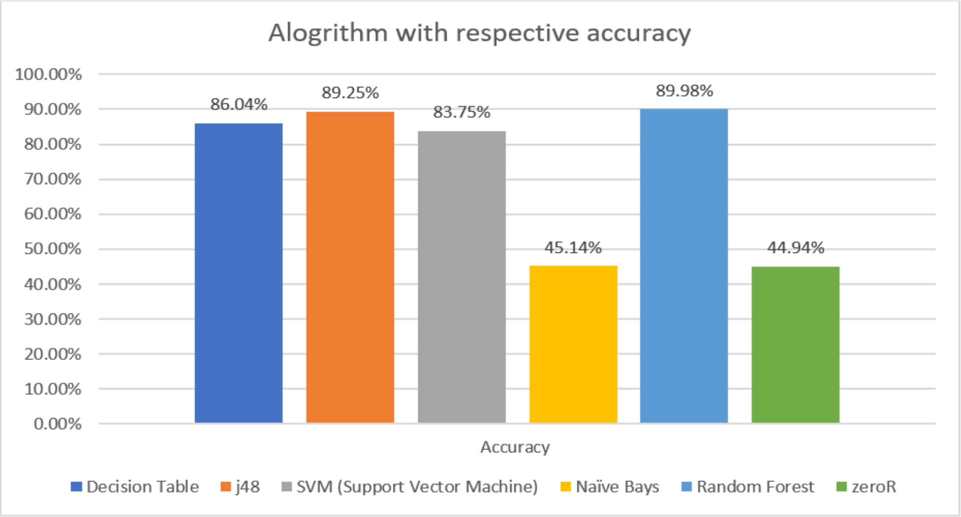


FIGURE 4.1: In Weka Algorithm Perfrmace with Repect to Accuracy

If we discuss the time with respect to accuracy of the algorithm that is so much important here because we are going to design the IDS for edge servers which usually have low resources so we need the algorithm which take the very less time to train and predict the result. So we analyze algorithms performance with respect to time. In below chart we also discuss it as well. We see that Random Forest gives the highest accuracy in 69 seconds whereas on second no j48 gives the second highest accuracy in 14 seconds and their gap of points in accuracy. So, we select the j48 algorithm here because this performs the very well in least amount of time. This experiment is just for the sake of motivation towards the actual experiments we discuss here it as an example how we perform the actual experiments and discuss their results.

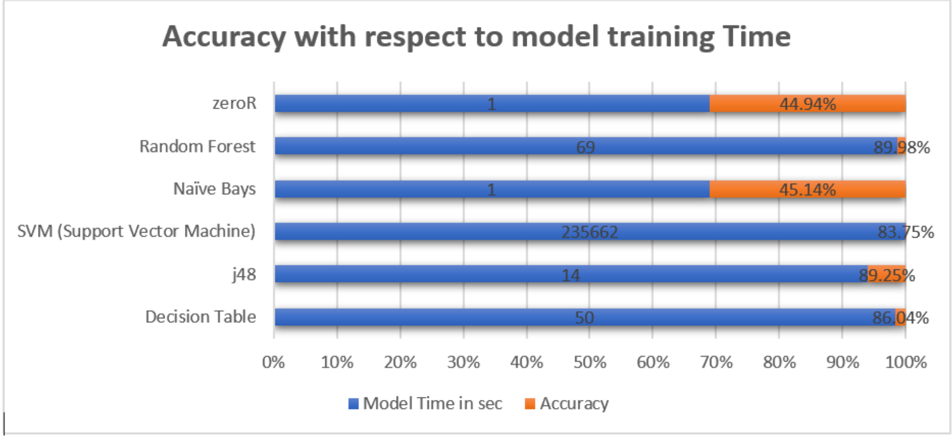


FIGURE 4.2: Weka algorithm accuracy with respect to time

4.3 Proposed Methodology Experiments

After the preprocessing on UNSW-NB15 dataset we will apply the selected algorithms on it and record the results. We also apply some setting on dataset which make it useable for machine learning algorithms.

The very firstly we read the dataset and verify dataset don't have null values. If we seem null value we remove it from the dataset or place the respective value for the location.

Because we are performing the multi class classification so we encode all the object type column values. For the reason we use the Label Encoder function available in the Sklearn library.

Then we separate the feature use for training and select the target attribute. Other input act as the input features and target attribute act as the resultant attribute.

After the above steps we separate the data to training and testing size. We use the 33% of data for the purpose of testing and remaining 67% for the training phase. We also use the random sate of 54% which shuffle the data before splitting to training and testing datasets. We have different files of training and testing dataset but when we use them separately, we seem the low accuracy because number of records in the training dataset are lower than in testing dataset. We will understand this below experiments. Here we done with the all basic configuration of dataset before passing them to actual machine learning algorithm.

4.3.1 Experiment 1

In our first experiment we pass the training dataset directly to selected machine learning algorithms with same code configurations we discuss in above para. After the processing we got the results sown in below table 4.1.

For more clarity we will discuss each column in graphical notation. The chart below shows the combine presentation of results shown in above table.

Table 4.1: Experiment 1 ML algorithm results with respect to time

Sr.	Algorithm	TTs	Af1	MAf1	WAf1	MAP	MAR	WAP	WAR
1.	Random Forest	15.83	89%	55%	89%	55%	54%	89%	89%
2.	Decision Tree	1.13	88%	57%	88%	53%	54%	88%	88%
3.	Neural Networks	12.07	45%	6%	28%	4%	10%	20%	45%
4.	Logistic Regression	4.78	61%	14%	50%	12%	18%	42%	61%
5.	Naïve Bays	0.30	41%	15%	40%	25%	20%	63%	41%
6.	SVM	658.53	55%	13%	45%	11%	17%	39%	55%

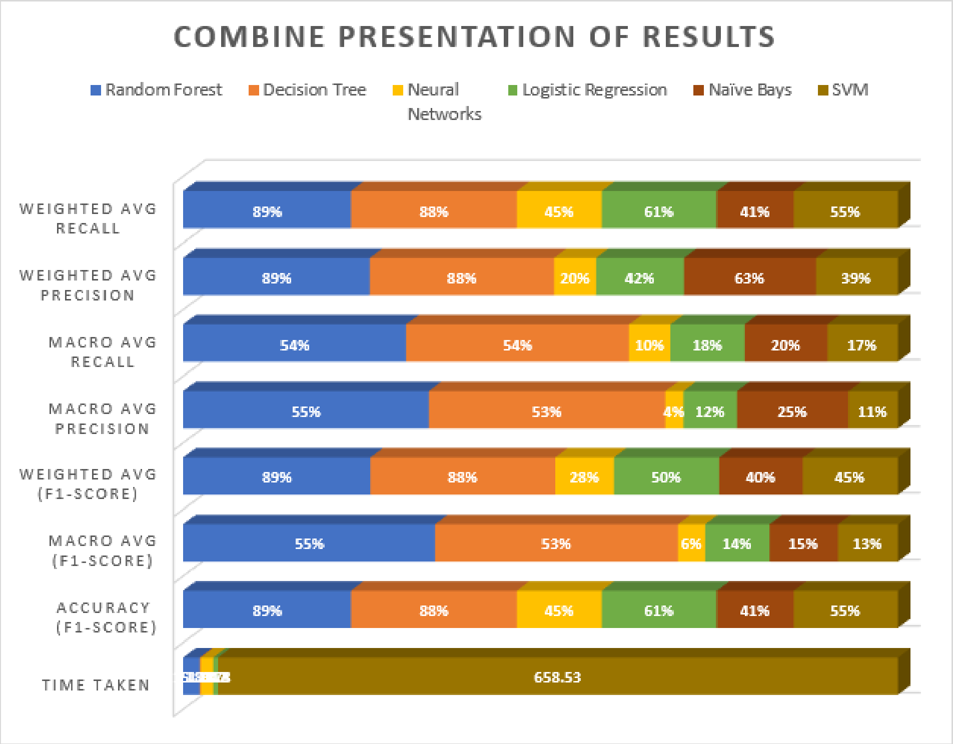


Figure 4.3: Experiment 1 Combine Presentation of Results

Above chart is little bit complex so we break each part of chart for more understandings. In below chart we show the time taken by each algorithm to build model and prediction of results.

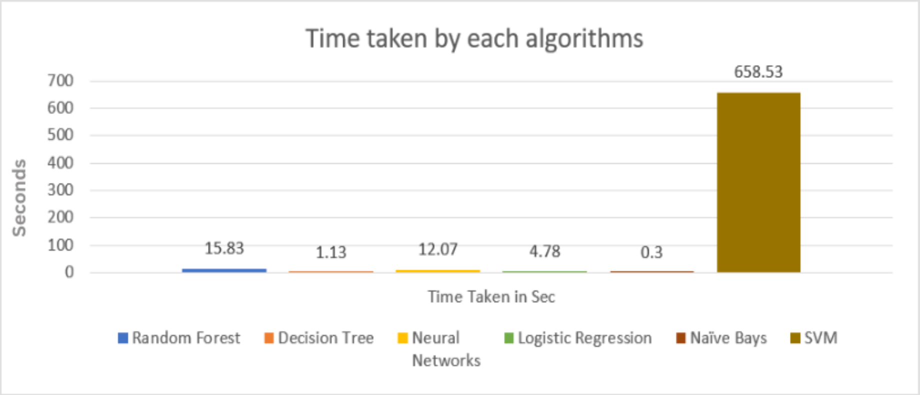


Figure 4.4: Experiment 1 time taken by each algorithm

Next we show the accuracy f1-score for each algorithm in below chart.

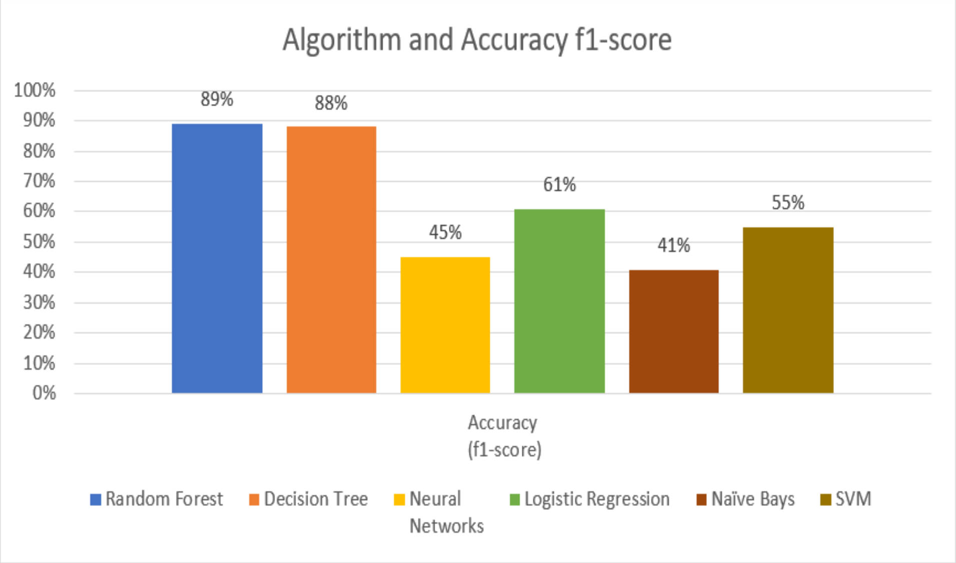


Figure 4.5: Experiment 1 Algorithms Accuracy f1 score

Macro avg and weighted avg of f1-score is shown below.

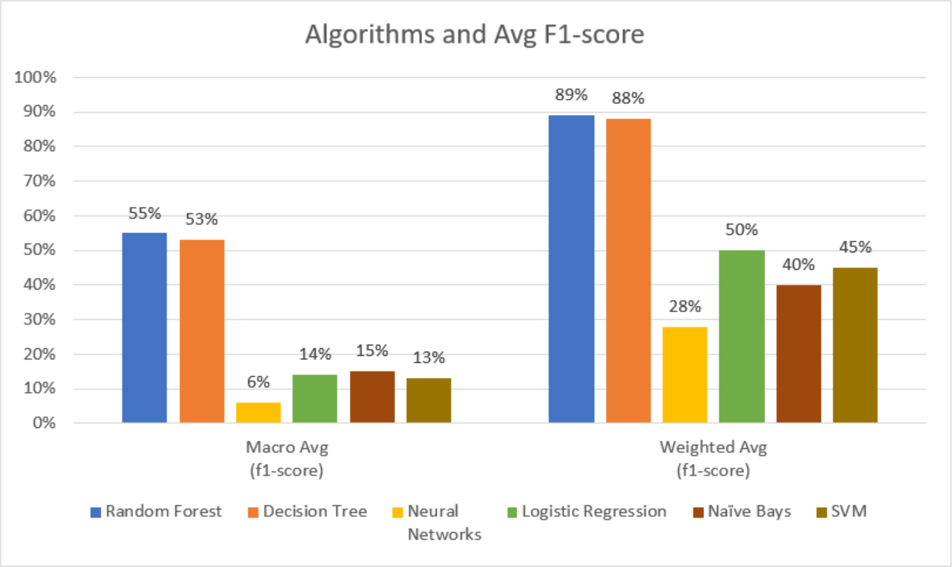


FIGURE 4.6: Experiment 1 Combine Macro and Weighted Avg f1 Score

Macro avg and weighted avg of Precision is shown below.

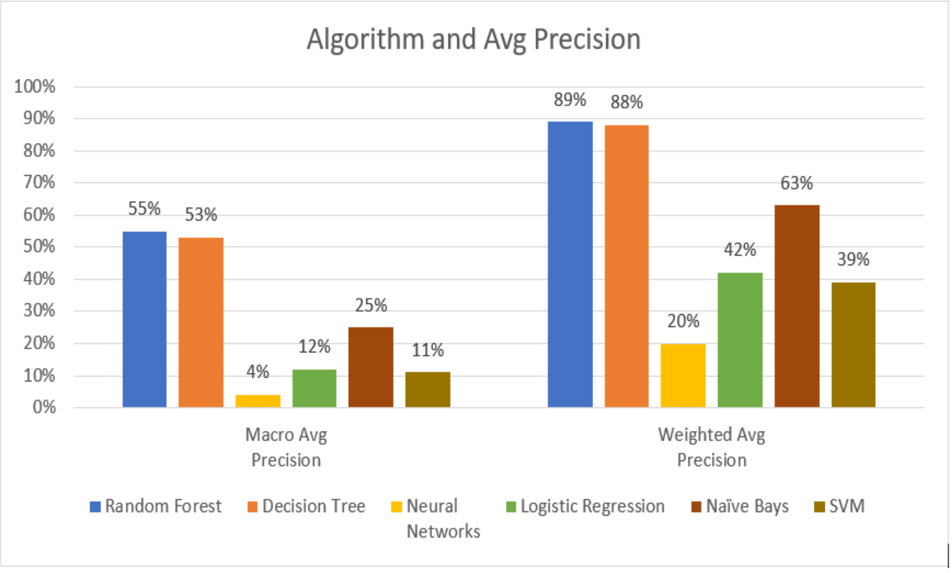


FIGURE 4.7: Experiment 1 Combine Macro and Weighted Avg of Precision

Macro avg and weighted avg of Recall is shown below.

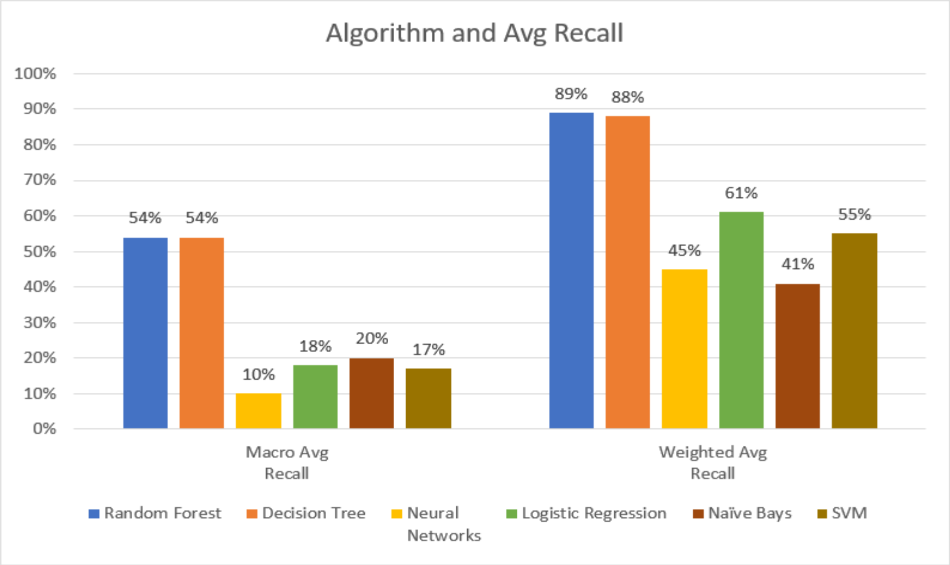


Figure 4.8: Experiment 1 Combine Macro and Weighted Avg of Recall

Because this is only the training dataset so it has the great performance in accuracy, f1-score, precision and recall. Due to the low number of rows in the dataset, the train model does not perform well in actual testing. Our next step is to conduct the actual experiments.

4.3.2 Experiment 2

Our second's experiment is based on the same above training dataset, we just drop the ID feature from the configuration because it is only use for the uniquely identification of records. The results are shown in below table 4.2.

Same as above for more clarity we will discuss each column in graphical notation. The chart below shows the combine presentation of results shown in above table. After Dropping ID feature accuracy of model gets decrease by respective algorithms. But in actual environment we don't have the id for the packet because those are real time packets so we eliminate this feature to make our results more realistic towards actual environment. As shown in Fig 4.9.

Time taken by each algorithm is shown below separately. SVM again takes the highest time and also not perform well for the dataset. As shown in Fig 4.10.

If we see the only F1- accuracy score for the model we see that Random Forest provides the highest accuracy than decision tree with a little bit difference. Graphical notation is shown in below Fig 4.11.

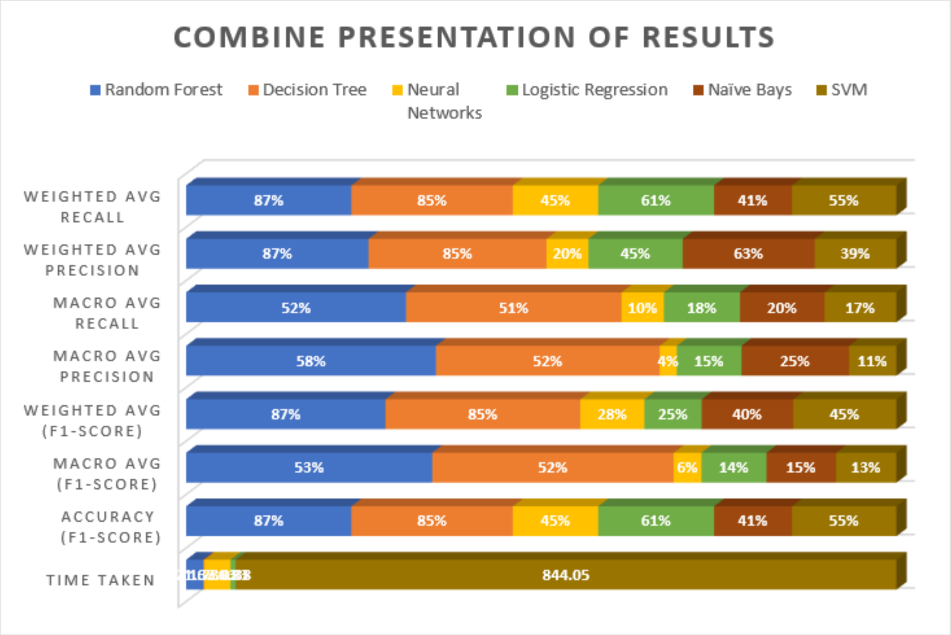


FIGURE 4.9: Experiment 2 Combine Presentation of Results

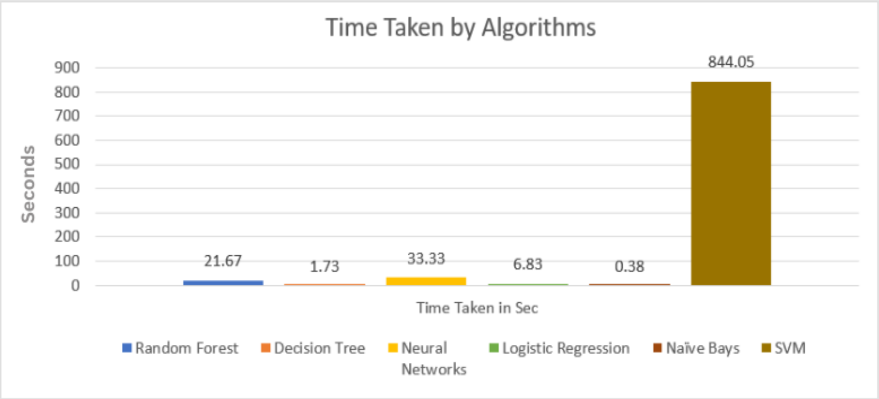


FIGURE 4.10: Experiment 2 time taken by each algorithm

Table 4.2: Experiment 2 ML algorithm results with respect to time

Sr.	Algorithm	TTs	Af1	MAf1	WAf1	MAP	MAR	WAP	WAR
1.	Random Forest	21.67	87%	53%	87%	58%	52%	87%	87%
2.	Decision Tree	1.73	85%	52%	85%	52%	51%	85%	85%
3.	Neural Networks	33.33	45%	6%	28%	4%	10%	20%	45%
4.	Logistic Regression	6.83	61%	14%	50%	15%	18%	45%	61%
5.	Naïve Bays	0.38	41%	15%	40%	25%	20%	63%	41%
6.	SVM	844.05	55%	13%	45%	11%	17%	39%	55%

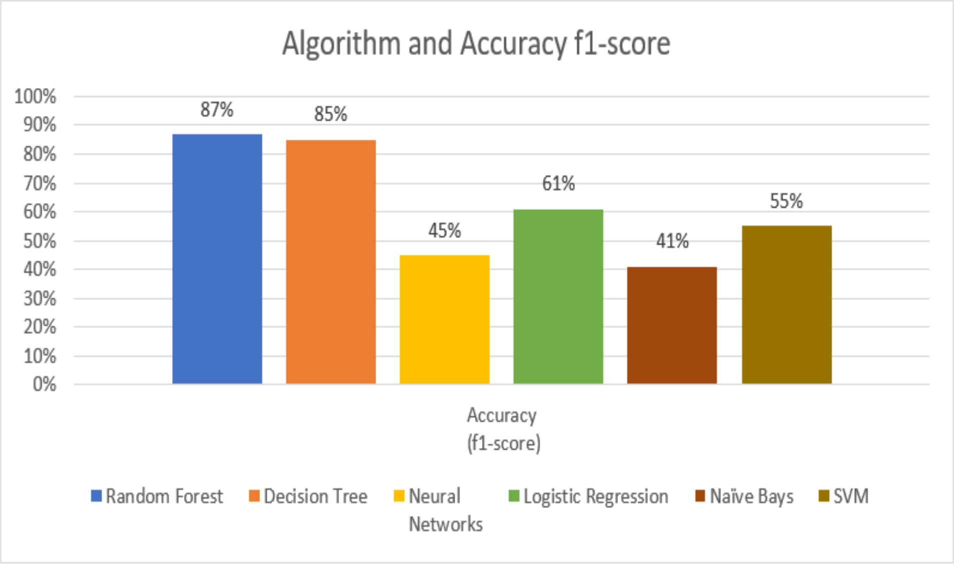


Figure 4.11: Experiment 2 Algorithms Accuracy f1 score

If we compare both Macro and Weighted accuracy we see that weighted avg is more batter than macro, because our dataset is also unbalanced so we have more trust on weighted accuracy. As shown in below Fig 4.12.

Macro avg and weighted avg of Precision is shown below Fig 4.13.

Macro avg and weighted avg of Recall is shown below Fig 4.14.

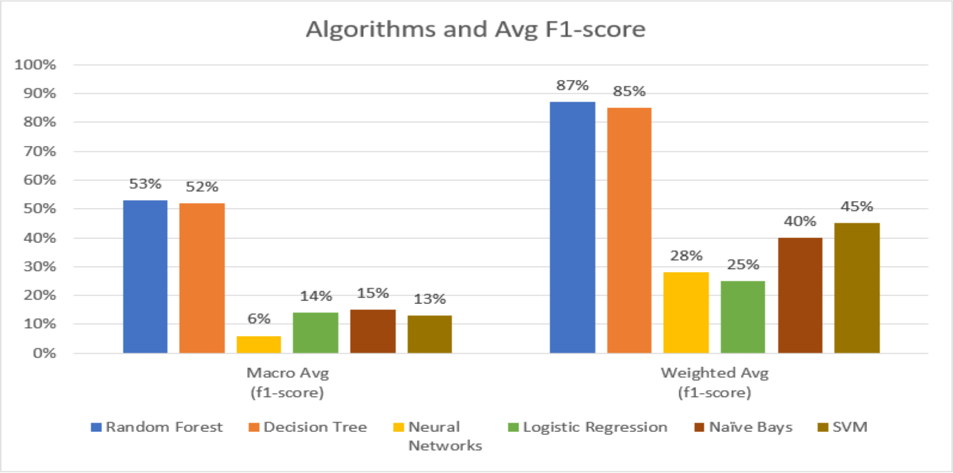


FIGURE 4.12: Experiment 2 Combine Macro and Weighted Avg f1 Score

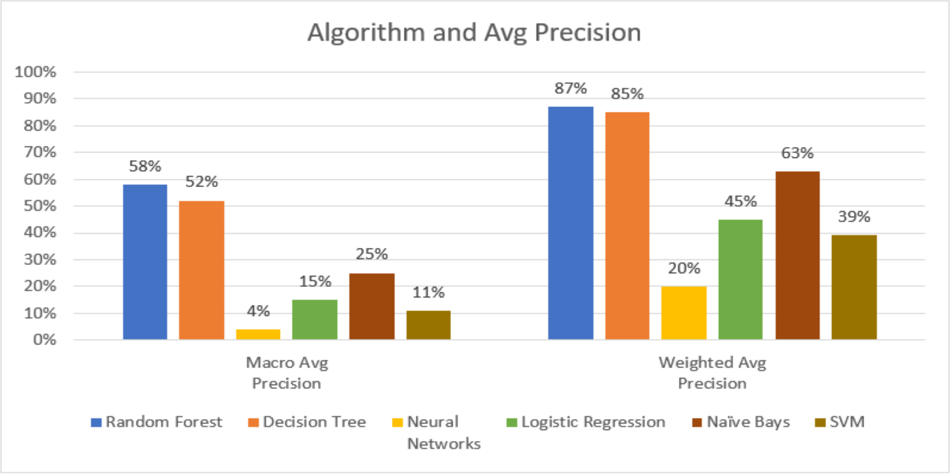


FIGURE 4.13: Experiment 2 Combine Macro and Weighted Avg of Precision

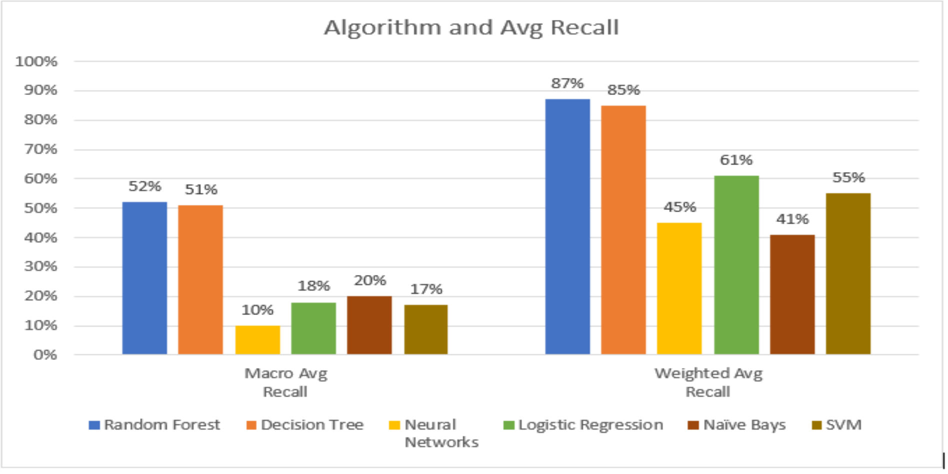


FIGURE 4.14: Experiment 2 Combine Macro and Weighted Avg of Recall

In this experiment we just delete the irrelevant feature from the dataset and perform the experiments but we change this only in the training dataset. We also perform the same experiment to select the best relevant experiment but we see that every feature is playing role in achieving accuracy we achieve the best possible accuracy only after utilizing all the features. We just ignore the ID feature because it is irrelevant and in realistic environment there will be no ID.

4.3.3 Experiment 3

In our third experiment we done with all possible cleaning mechanisms we just train model and test it on respective test file. Obtain results are discussed below in table 4.13.

Same as above for more clarity we will discuss each column in graphical notation. The chart below shows the combine presentation of results shown in above table. We see the huge decrease in the accuracy in any terms because training data is lesser than the testing dataset. Both files have the same features but only the difference in number of records. So the reason is that we have less data to train our model so the performance of the model is not so well as shown below in Fig 4.15.

TABLE 4.3: Experiment 3 ML algorithm results with respect to time

Sr. #	Algorithm	TTs	Af1	MAf1	WAf1	MAP	MAR	WAP	WAR
1.	Random Forest	60	73%	47%	70%	57%	45%	72%	73%
2.	Decision Tree	1.67	70%	53%	68%	61%	53%	71%	70%
3.	Neural Networks	11.86	30%	5%	14%	3%	10%	9%	30%
4.	Logistic Regression	6.05	49%	13%	34%	20%	19%	37%	49%
5.	Naïve Bays	11	43%	18%	39%	28%	22%	58%	43%
6.	SVM	935.47	42%	11%	29%	8%	17%	22%	42%

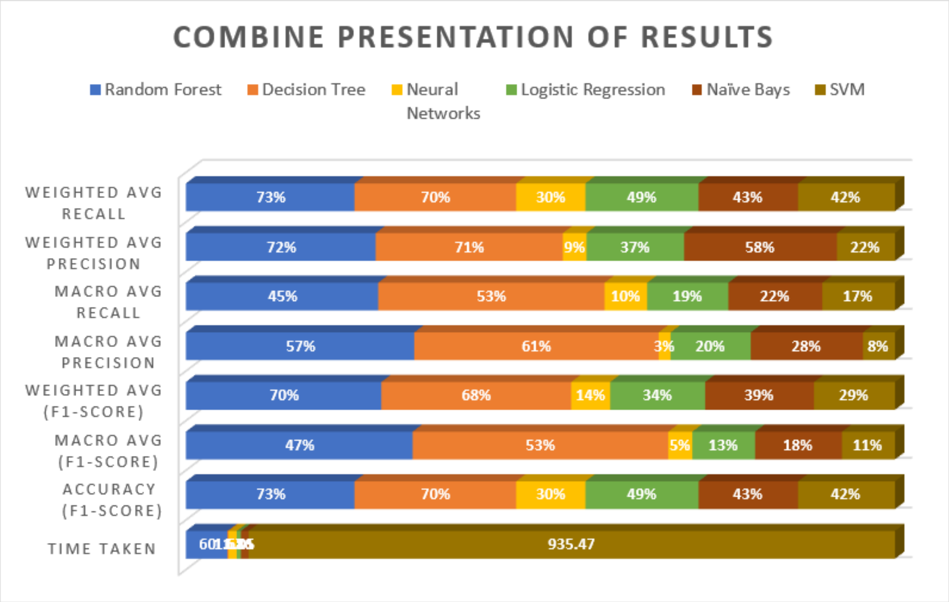


FIGURE 4.15: Experiment 3 Combine Presentation of Results

Time taken by each algorithm is shown below separately. SVM again takes the highest time and also not perform well for the dataset.

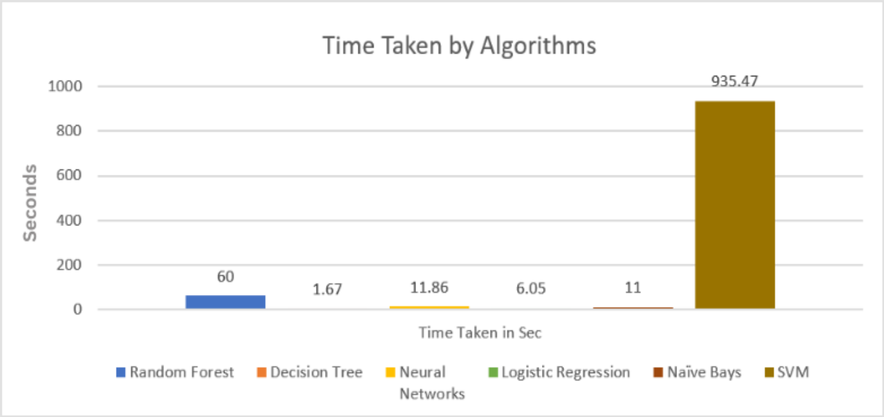


FIGURE 4.16: Experiment 3 time taken by each algorithm

If we see the only F1- accuracy score for the model we see that Random Forest provides the highest accuracy than decision tree with a little bit difference. Graphical notation is shown below Fig 4.17

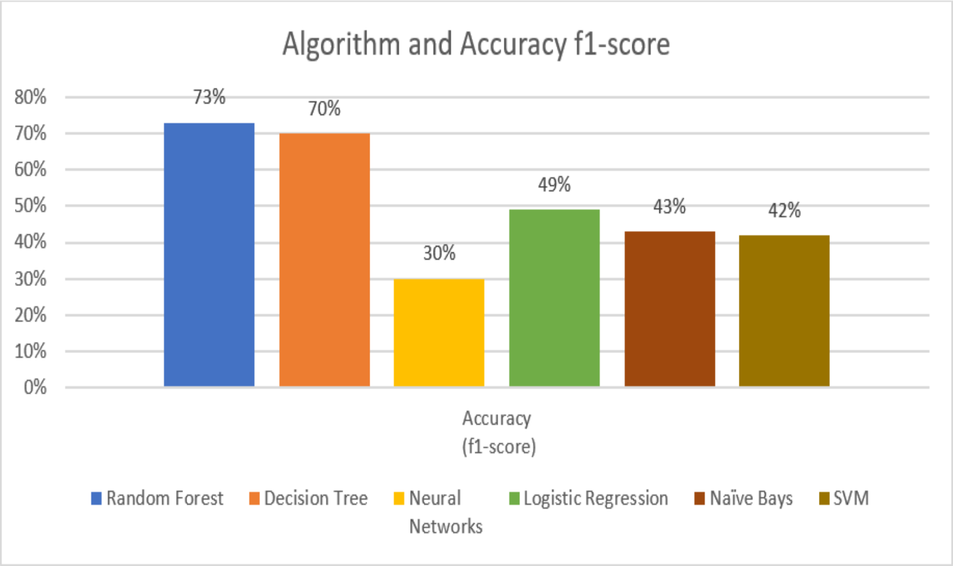


FIGURE 4.17: Experiment 3 Algorithms Accuracy f1 score

If we compare both Macro and Weighted accuracy we see that weighted avg is more batter than macro, because our dataset is also unbalanced so we have more trust on weighted accuracy. So weighted accuracy of Random Forest is batter.

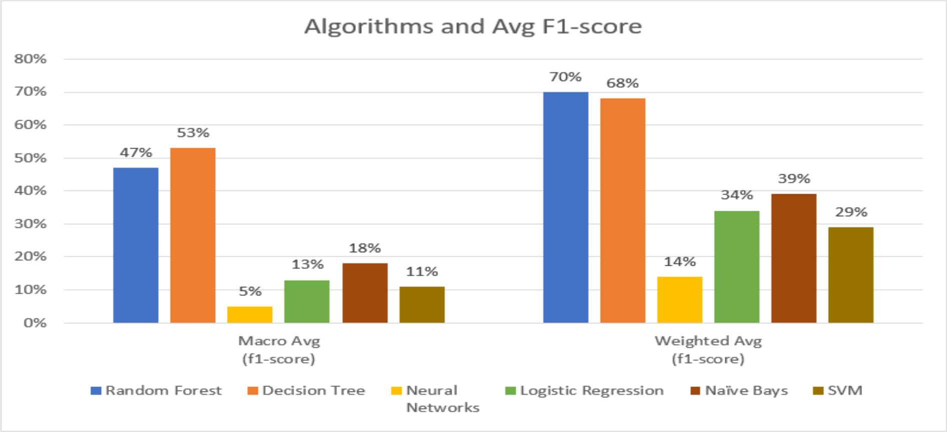


FIGURE 4.18: Experiment 3 Combine Macro and Weighted Avg f1 Score

Macro avg and weighted avg of Precision is shown below Fig 4.19.

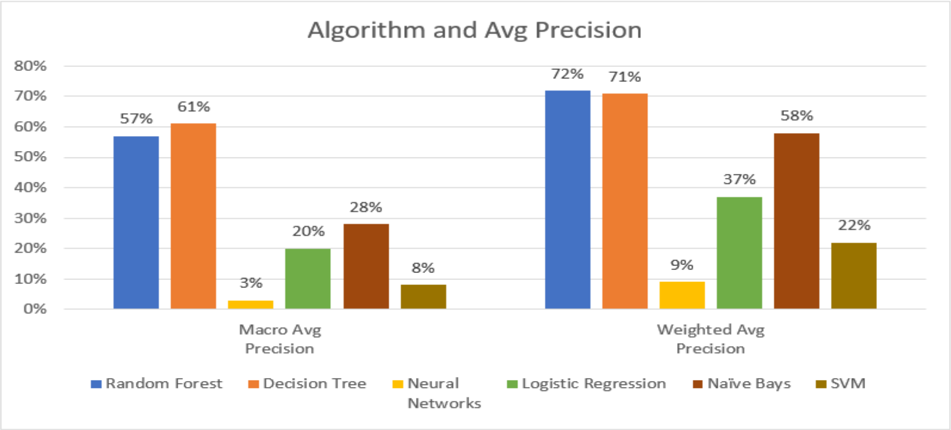


FIGURE 4.19: Experiment 3 Combine Macro and Weighted Avg of Precision

Macro avg and weighted avg of Recall is shown below 4.20.

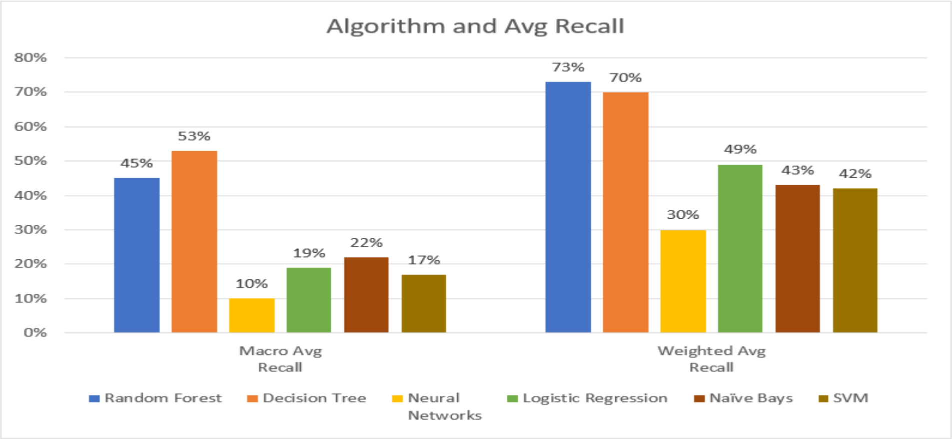


FIGURE 4.20: Experiment 3 Combine Macro and Weighted Avg of Recall

We see that all algorithms decrease their accuracy because in accuracy dataset also plays an important role. Here we train our model on small training dataset and in comparison, we have large testing data which cause the decrease in actual accuracy. Because testing data also have the same features so we combine both in training and testing in our next experiment.

4.3.4 Experiment 4

In our fourth experiment we combine the both training and testing dataset into single file and then utilized the 33% for testing and remaining for training. The statistics of results are shown in table 4.14 The obtain results looks much batter as shown below

For understanding and clarity below we show the data in the charts. The combine presentation of results shown below Fig 4.21

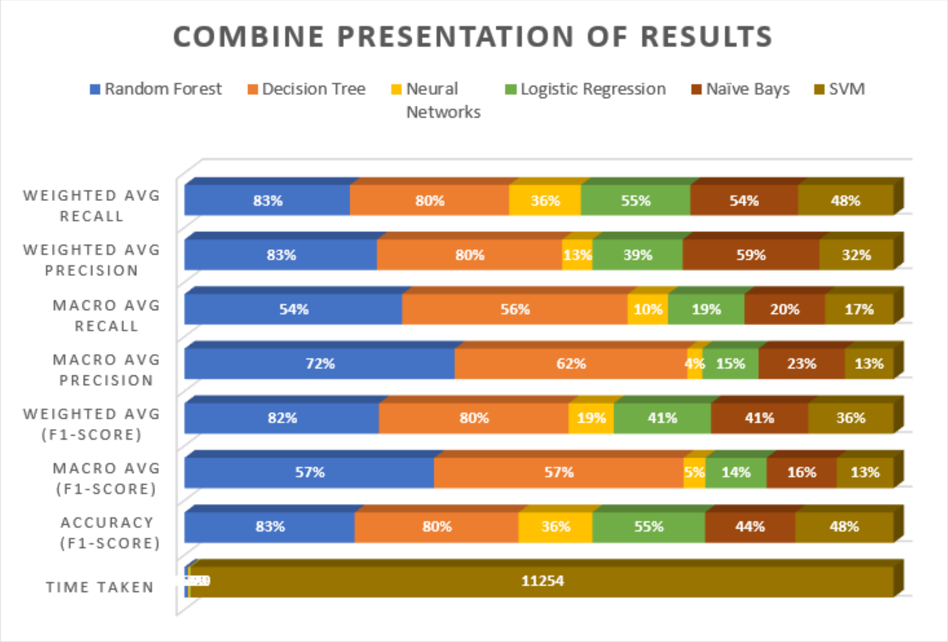


FIGURE 4.21: Experiment 4 Combine Presentation of Results

Time take by each algorithm is shown below again svm takes the highest amount of time. Fig 4.22

Table 4.4: Experiment 4 ML algorithm results with respect to time

Sr. #	Algorithm	TTs	Af1	MAf1	WAf1	MAP	MAR	WAP	WAR
1.	Random Forest	56.65	83%	57%	82%	72%	54%	83%	83%
2.	Decision Tree	4.81	80%	57%	80%	62%	56%	80%	80%
3.	Neural Networks	26.20	36%	5%	19%	4%	10%	13%	36%
4.	Logistic Regression	16.29	55%	14%	41%	15%	19%	39%	55%
5.	Naïve Bays	0.92	44%	16%	41%	23%	20%	59%	54%
6.	SVM	1125.40	48%	13%	36%	13%	17%	32%	48%

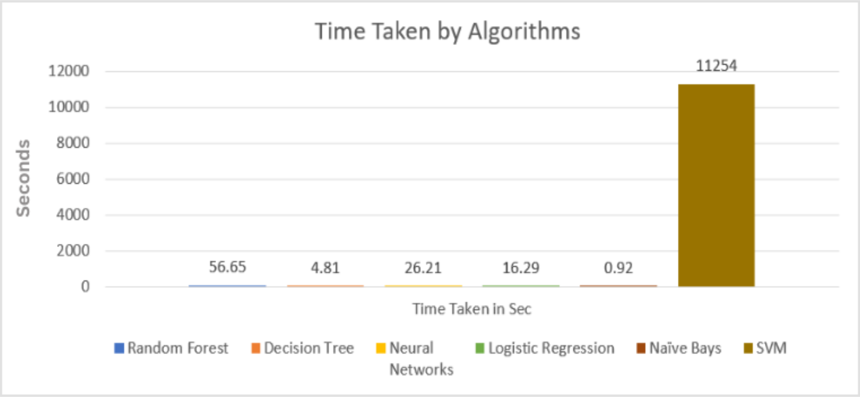


Figure 4.22: Experiment 4 time taken by each algorithm

If we see the only F1- accuracy score for the model we see that Random Forest provides the highest accuracy than decision tree with a little bit difference. Graphical notation is shown below Fig 4.23

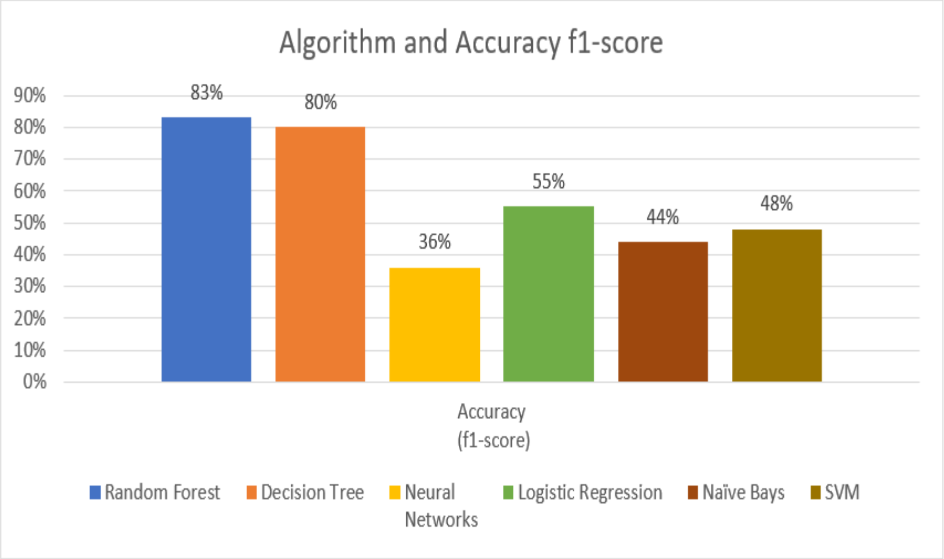


FIGURE 4.23: Experiment 4 Algorithms Accuracy f1 score

If we compare both Macro and Weighted accuracy, we see that weighted avg is more batter than macro, because our dataset is also unbalanced so we have more trust on weighted accuracy. So weighted accuracy of Random Forest is batter.

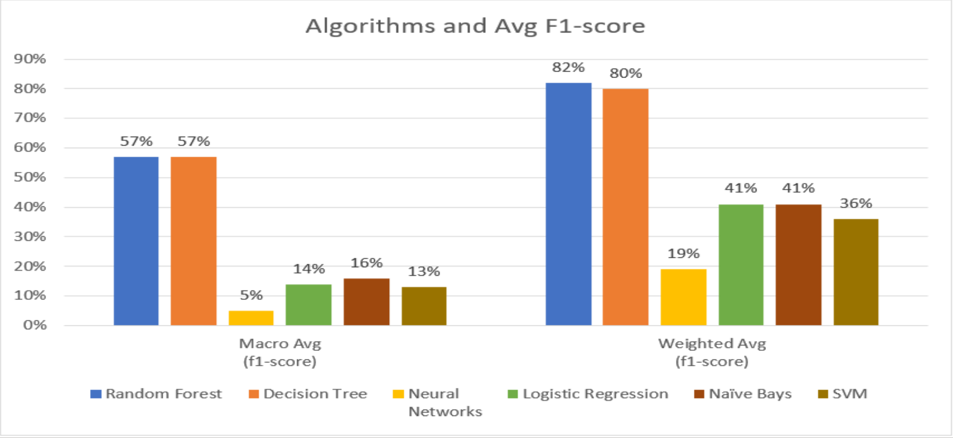


FIGURE 4.24: Experiment 4 Combine Macro and Weighted Avg f1 Score

Macro avg and weighted avg of Precision is shown below.

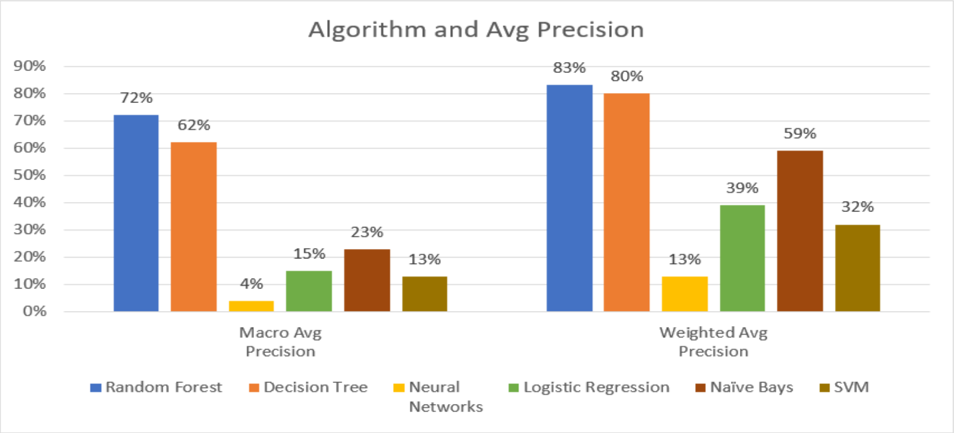


Figure 4.25: Experiment 4 Combine Macro and Weighted Avg of Precision

Macro avg and weighted avg of Recall is shown below.

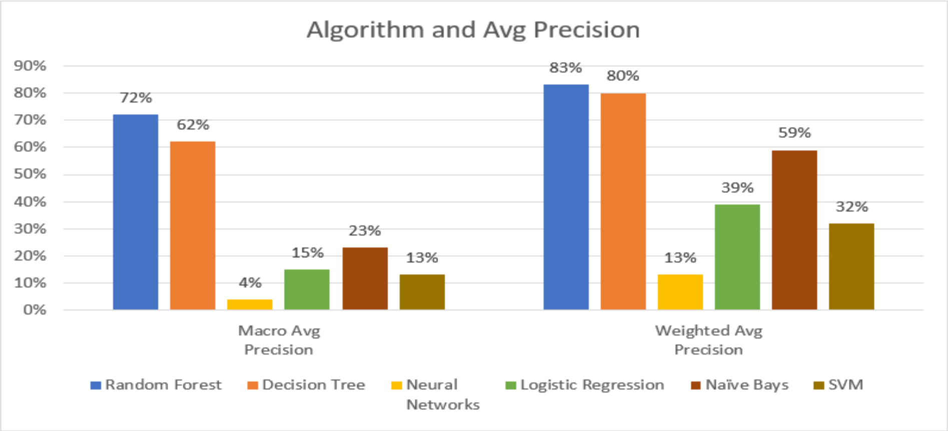


Figure 4.26: Experiment 4 Combine Macro and Weighted Avg of Recall

We see the increase in accuracy, precision and in recall when we combine the both training and testing dataset, this is because number of records for the training model gets increase which cause the good model building and model perform well as compare to previous one.

4.3.5 Experiment 5

In our fifth experiment we combine the both training and testing dataset into single file and then utilized the 33% for testing and remaining for training as in experiment 4. In addition, we sort out the data by attack class feature than we train the model and we see the increase in the accuracy. We also use the KNN algorithm for value 3 in this experiment but KNN did not play well so we ignore this in above experiments. The obtain results looks much batter as shown below table 4.5

For understanding and clarity below we show the results presentation in Fig4.27.

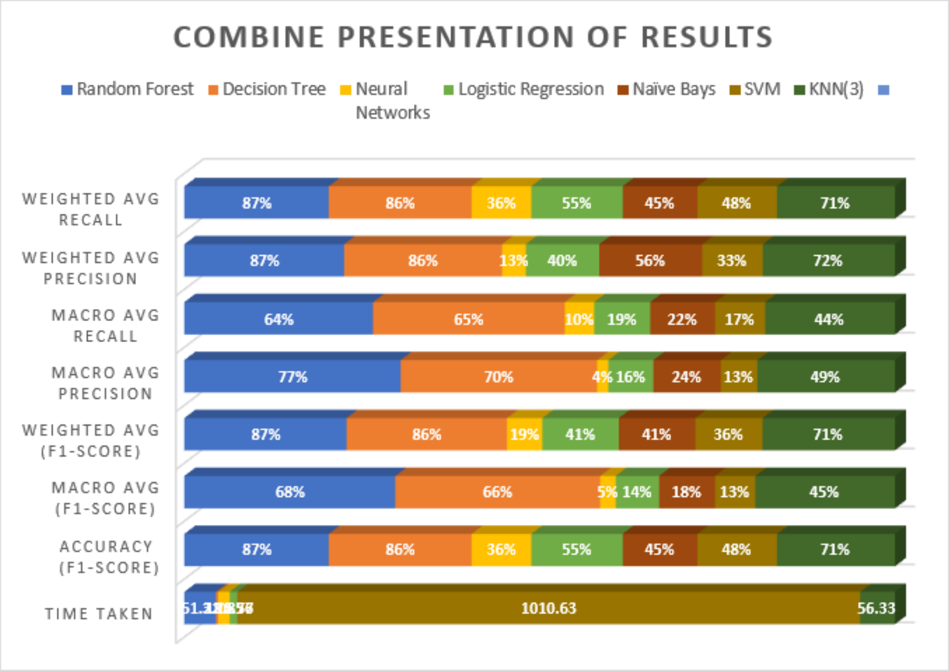


FIGURE 4.27: Experiment 5 Combine Presentation of Results

Time take by each algorithm is shown below again SVM takes the highest amount of time.

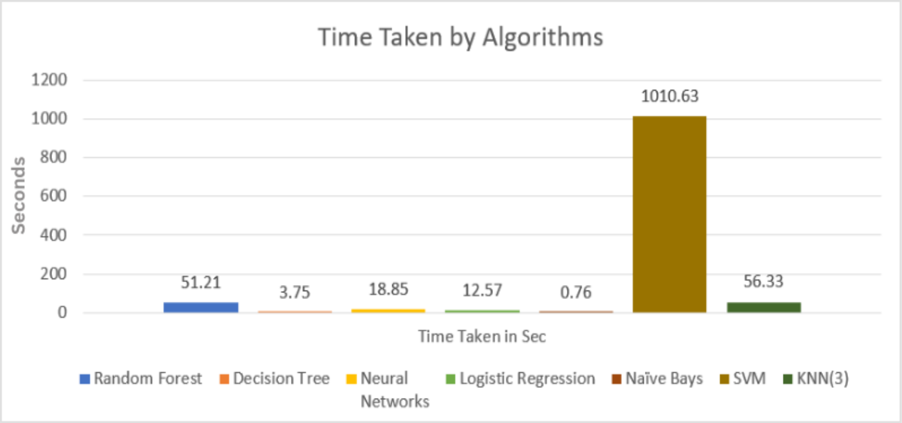


FIGURE 4.28: Experiment 5 time taken by each algorithm

Table 4.5: Experiment 5 ML algorithm results with respect to time

Sr. #	Algorithm	TTs	Af1	MAf1	WAf1	MAP	MAR	WAP	WAR
1.	Random Forest	51.20	87%	68%	87%	77%	64%	87%	87%
2.	Decision Tree	3.75	86%	66%	86%	70%	65%	86%	86%
3.	Neural Net-works	18.85	36%	5%	19%	4%	10%	13%	36%
4.	Logistic Re-gres-sion	12.57	55%	14%	41%	16%	19%	40%	55%
5.	Naïve Bays	0.76	45%	18%	41%	24%	22%	56%	45%
6.	SVM	1010.63	48%	13%	36%	13%	17%	33%	48%
7.	KNN(3)	56.33	71%	45%	71%	49%	44%	72%	71%

If we see the only F1- accuracy score for the model we see that Random Forest provides the highest accuracy than decision tree with a little bit difference. Graphical notation is shown below Fig 4.29

If we compare both Macro and Weighted accuracy, we see that weighted avg is more batter than macro, because our dataset is also unbalanced so we have more trust on weighted accuracy. So weighted accuracy of Random Forest is batter.

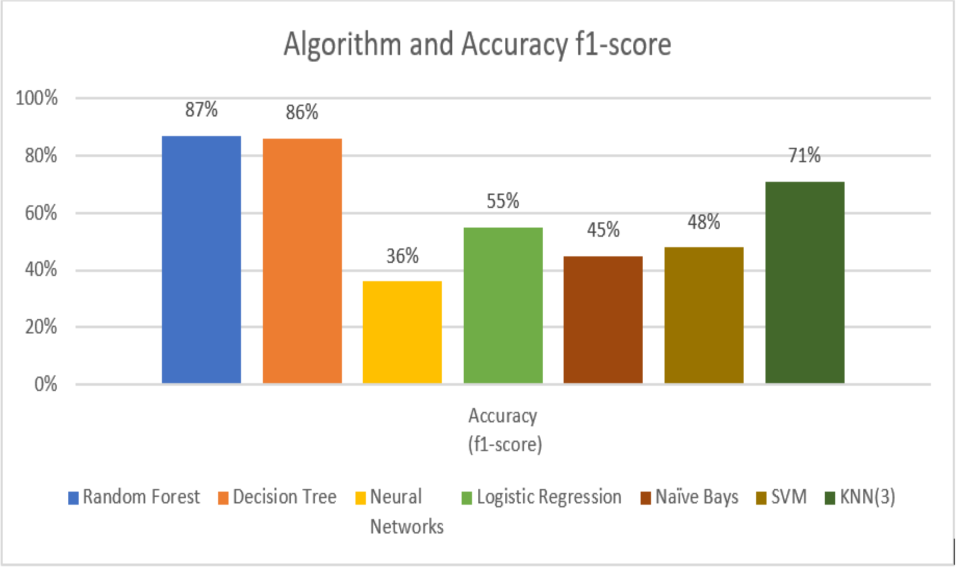


Figure 4.29: Experiment 5 Algorithms Accuracy f1 score

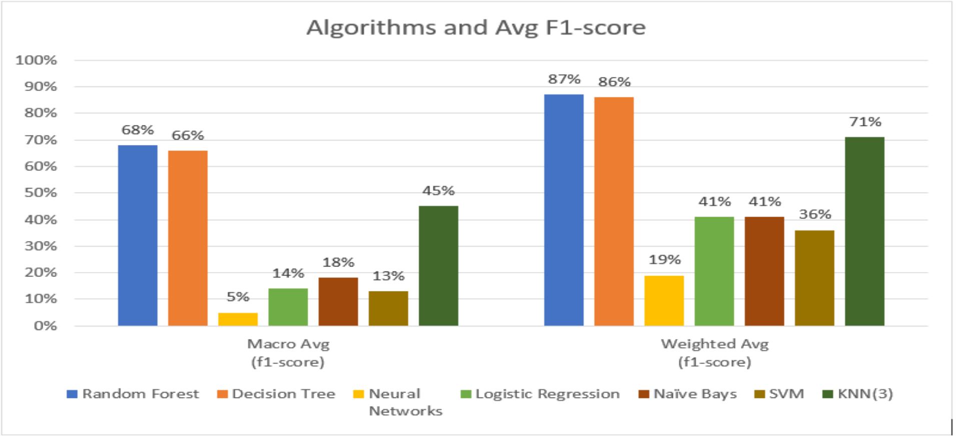


Figure 4.30: Experiment 5 Combine Macro and Weighted Avg f1 Score

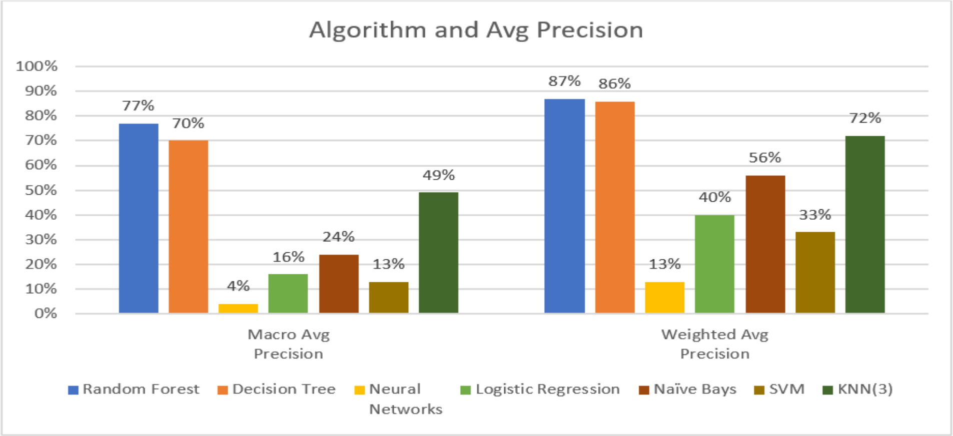


Figure 4.31: Experiment 5 Combine Macro and Weighted Avg of Precision

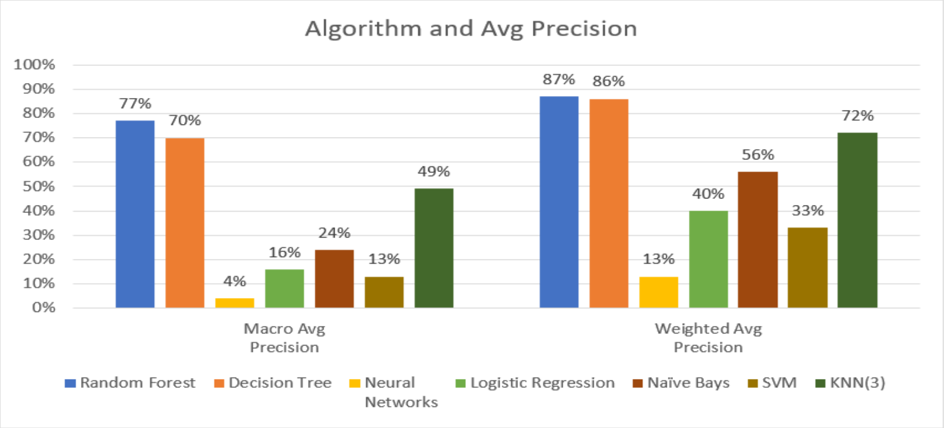


Figure 4.32: Experiment 5 Combine Macro and Weighted Avg of Recall

Macro avg and weighted avg of Precision is shown above Fig 4.31 and Macro avg and weighted avg of Recall is shown above Fig 4.32.

We see improvements in result because we resample the dataset. On the bases of obtain results we see that Random Forest and Decision Tree Plays very well in term of accuracy, precision and recall. There is only 1 point difference is seen which is ignore able when we compare its results with the time. Decision Tree is light weight and takes second lowest time of 3.75 seconds where as Random Forest takes 51.21 seconds. Here we just discuss the results but in next chapter we will high light the preferred algorithm which we will proposed for use in NIDS engine with corresponding dataset.

4.3.6 Experiment 6

Experiment 1 to 5 are related to NIDS engine so we used the machine learning algorithms and analyze the performance of each experiment in respective of algorithm. This experiment is related to SIDS engine which are being programmed in python programming language. This engine receives the run time mirror traffic from the sensors and we define some rules for each sensor. When there is https we have to add the https certificate which cause the decryption of traffic for particular sensor. SIDS engine simply inspect the received traffic on bases of define rules if there is any miss match than engine hit the respective alert. This also monitor the connection state of sensors if any sensor loss the connection than it also hits an alert. This engine also track the sensors data send if sensor not sends data in define time than system check the sensor state by ping and generate alert in case of intrusion. Below we show some SIDS engine alerts

Logs

```
Got connection from ('192.168.31.1', 63912)
['Blood Sensor', 'int', '60', '100']
['90', '50000'] 2
Intrusion Data −− > Sensor: Blood Sensor is out of Range value: 50000
Got connection from ('192.168.31.1', 63914)
['Glucose Sensor', 'int', '60', '100']
['90', '50000'] 2
Intrusion Data −− > Sensor: Glucose Sensor is out of Range value: 50000
['90', '50000'] 2
Intrusion Data −− > Sensor: Glucose Sensor is out of Range value: 50000
Intrusion Data −− > Sensor: Blood Sensor is not Sending Updated Data since
10.13837718963623
Intrusion Data −− > Sensor: Blood Sensor is not Sending Updated Data since
11.142565965652466
Intrusion Data −− > Sensor: Blood Sensor is not Sending Updated Data since
12.147562980651855
['90', '50000'] 2
Intrusion Data −− > Sensor: Glucose Sensor is out of Range value: 50000
Intrusion Data −− > Sensor: Blood Sensor is not Sending Updated Data since
13.161430835723877
Intrusion Data −− > Sensor: Blood Sensor is not Sending Updated Data since
14.164147138595581
Intrusion Data −− > Sensor: Glucose Sensor has Problem in sending Data MSG
class 'ConnectionResetError
sensor Lost Connection
['90', '50000'] 2
```

```
Intrusion Data −− > Sensor: Blood Sensor is out of Range value: 50000
Intrusion Data −− > Sensor: Glucose Sensor is lost connection
Intrusion Data −− > Sensor: Blood Sensor has Problem in sending Data MSG
'ConnectionResetError'
sensor Lost Connection
Intrusion Data −− > Sensor: Blood Sensor is lost connection
Intrusion Data −− > Sensor: Glucose Sensor is lost connection
Intrusion Data −− > Sensor: Blood Sensor is lost connection
Intrusion Data −− > Sensor: Glucose Sensor is lost connection
```

4.4 Evaluation

After conducting each experiment, we evaluate it in terms of accuracy, Macro Avg (f1-score, Precision, Recall) and Weighted Avg (f1-score, Precision, Recall). Our major parameter is Weighted Avg because we used the unbalanced dataset so for each attack class we have different number of records. When we compare the two best algorithms result of each experiment, we see that mostly Random Forest and Decision Tree Plays best in the minimum amount of time. Experiment 1 just have the training dataset so its model is just train over the train data and also test on train dataset so we have the high accuracy but this is not performed well in the actual environment so we perform different operations on the dataset and extract the different results. We see that only Random Forest and Decision Tress achieve the high accuracy with little bit difference. With complete dataset we achieve the highest accuracy in the experiment 5 which model is good to detect the actual environment attacks because it is train and test over the more data. Because we are on edge computing so there is possibility of low resources so according to performance with respect to time decision tree is the best algorithm to use with UNSB-NB 15 dataset with define properties. It takes the least amount of time and perform the very well and we achieve the accuracy of 86% with weighted f1-score, precision and recall. So we recommend to Decision Tree Algorithm for building NIDS engine.

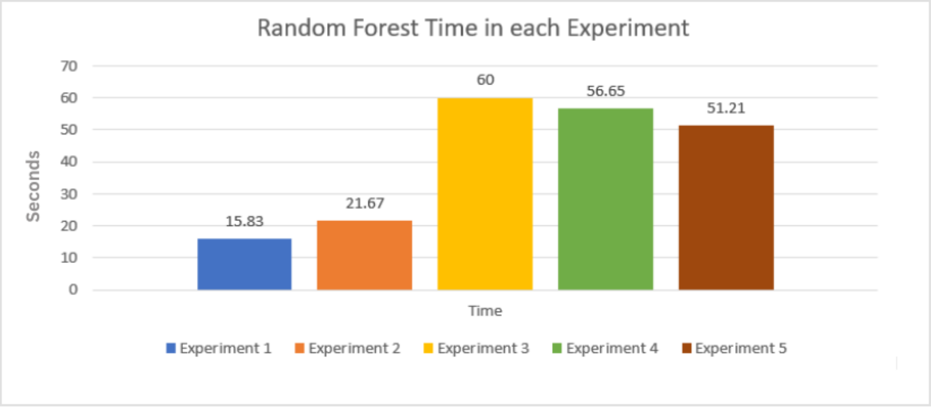


FIGURE 4.33: Random Forest time in each Experiment

In Fig 4.33 we see the time takes by the Random Forest in each experiment and In Fig 4.34 we see Results of Random Forest in each experiment.

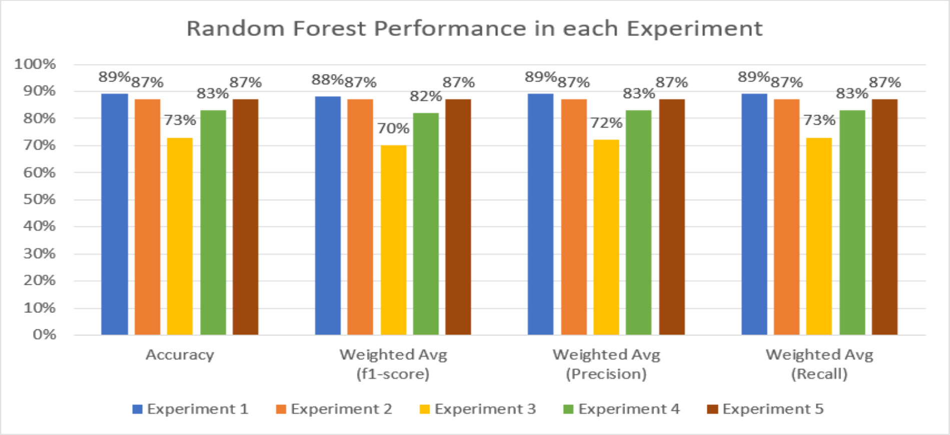


FIGURE 4.34: Random Forest Performance in each Experiment

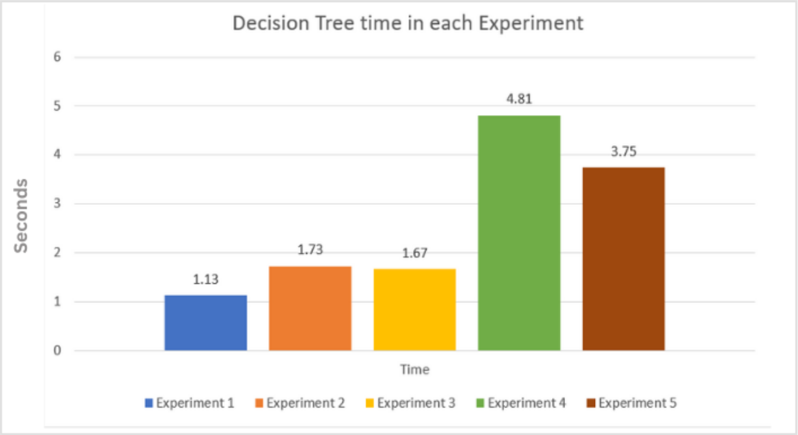


FIGURE 4.35: Decision Tree time in each Experiment

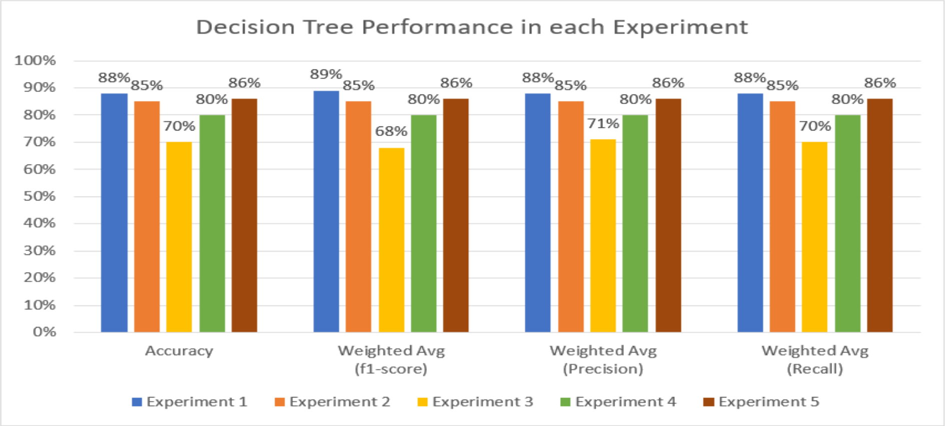


FIGURE 4.36: Decision Tree Performance in each Experiment

In Fig 3.5 we see the time takes by the Decision Tree in each experiment and In Fig 3.6 we see Results of Decision Tree in each experiment.

In next Chapter will discuss the candidate selected algorithm and reason behaind why we select it.

Chapter 5

Conclusion and Future work

Cyber warfare is the fifth generation of warfare that does not have borders and cannot be seen by the naked eye. Furthermore, these attacks are dangerous since we do not need to have a complete operation or force inside the enemy's country to succeed. An individual from another country could be responsible for heavy losses. Research work found that today's control systems utilize the internet of things (IoT), which poses some security risks because of their direct connection to the internet. Because of these risks, several different research questions need to be addressed immediately. Our research work, as noted in the introduction, answered some questions regarding these problems, as well as the design of an IDS that is suitable for edge computing architecture to protect against network attacks and alert for sensor intrusions. Additionally, research work also designed an IDS to protect against network attacks and alert for sensor intrusions.

As a result of studying comprehensive research surveys, discussions, and articles, we provide our research methodology in response to the research questions. We also provide a complete architecture of IDS for edge computing. IDS is further divided into NIDS and SIDS. In order to build a machine learning model for the detection of modern attacks, research work conducts the complete process of selecting the dataset. After the comparisons with different datasets, research work found the UNSB-NB 15 as the benchmark dataset then we find the methodology in [2] in which the author achieves the accuracy of 86% by layered model. Research work improves the model and achieves the same level of accuracy without a layered approach. Research work analyzes the dataset over 7 different machine learning algorithms Random Forest play the best in term of accuracy weighted (f1-score, precision, and recall) but it takes time whereas Decision Tree also performs best with 1% less accuracy than Random Forest and takes least amount of time. So, we finalize the Decision Tree Algorithm, for the NIDS engine to use with UNSB-NB 15 Dataset. As part of NIDS, we highlight the use of signatures with anomaly-based engines which enables early detection of confirmed attacks. As for SIDS, we only provide the programming method to program SIDS that detects sensor intrusions and also provide its prototype whose results are presented in experiment 6.

A future focus of our research is to develop different edge computing architectures using the SIDS and NIDS engines in the context of edge servers. However, we tuned the edge server as light as possible, so we need to test those on different edge computing concepts (like raspberry pi, PC, or server as an edge) so that we can see how well the proposed architecture performs.

Bibliography

[1] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: State of the art and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1608–1631, 2019.

[2] S. Meftah, T. Rachidi, and N. Assem, "Network based intrusion detection using the unsw-nb15 dataset," *International Journal of Computing and Digital Systems*, vol. 8, no. 5, pp. 478–487, 2019.

[3] S. Balaji, K. Nathani, and R. Santhakumar, "Iot technology, applications and challenges: a contemporary survey," *Wireless personal communications*, vol. 108, no. 1, pp. 363–388, 2019.

[4] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[5] F. Khan, M. A. Jan, A. ur Rehman, S. Mastorakis, M. Alazab, and P. Watters, "A secured and intelligent communication scheme for iiot-enabled pervasive edge computing," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5128–5137, 2020.

[6] I. U. Din, M. Guizani, B.-S. Kim, S. Hassan, and M. K. Khan, "Trust management techniques for the internet of things: A survey," *IEEE Access*, vol. 7, pp. 29763–29787, 2018.

[7] T. Gebremichael, L. P. Ledwaba, M. H. Eldefrawy, G. P. Hancke, N. Pereira, M. Gidlund, and J. Akerberg, "Security and privacy in the industrial internet

of things: Current standards and future challenges," *IEEE Access*, vol. 8, pp. 152351–152366, 2020.

[8] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "A trustworthy privacy preserving framework for machine learning in industrial iot systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6092–6102, 2020.

[9] D. Resul and M. Z. Gündüz, "Analysis of cyber-attacks in iot-based critical infrastructures," *International Journal of Information Security Science*, vol. 8, no. 4, pp. 122–133, 2020.

[10] C.-Y. Weng, C.-T. Li, C.-L. Chen, C.-C. Lee, and Y.-Y. Deng, "A lightweight anonymous authentication and secure communication scheme for fog computing services," *IEEE Access*, vol. 9, pp. 145522–145537, 2021.

[11] J. Li, J. Cai, F. Khan, A. U. Rehman, V. Balasubramaniam, J. Sun, and P. Venu, "A secured framework for sdn-based edge computing in iot-enabled healthcare system," *IEEE Access*, vol. 8, pp. 135479–135490, 2020.

[12] T. T. Huong, T. P. Bac, D. M. Long, B. D. Thang, N. T. Binh, T. D. Luong, and T. K. Phuc, "Lockedge: Low-complexity cyberattack detection in iot edge computing," *IEEE Access*, vol. 9, pp. 29696–29710, 2021.

[13] M. Yamauchi, Y. Ohsita, M. Murata, K. Ueda, and Y. Kato, "Anomaly detection in smart home operation from user behaviors and home conditions," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 2, pp. 183–192, 2020.

[14] K. Sadiq, A. Thompson, and A. Ayeni, "Mitigating ddos attacks in cloud network using fog and sdn: A conceptual security framework," *International Journal of Applied Information Systems*, vol. 12, no. 32, pp. 11–16, 2020.

[15] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," *IEEE Access*, vol. 8, pp. 106576–106584, 2020.

[16] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[17] F. A. Bakhtiar, E. S. Pramukantoro, and H. Nihri, "A lightweight ids based on j48 algorithm for detecting dos attacks on iot middleware," in *2019 IEEE 1st Global Conference on Life Sciences and Technologies (LifeTech)*, pp. 41–42, IEEE, 2019.

[18] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban ids: An intelligent anomaly-based intrusion detection system for iot edge devices," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, 2020.

[19] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9395–9409, 2022.

[20] M. Ghurab, G. Gaphari, F. Alshami, R. Alshamy, and S. Othman, "A detailed analysis of benchmark datasets for network intrusion detection system," *Asian Journal of Research in Computer Science*, vol. 7, no. 4, pp. 14–33, 2021.

[21] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18–31, 2016.

[22] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, pp. 1–6, IEEE, 2015.